OneOS V4.3R4

OneOS V5.1R3

WAN User Guide

(edition 10)

OneAccess Networks
Pentagone Plaza
381, Avenue du Général de Gaulle
92140 Clamart
FRANCE

The law of 11 March 1957, paragraphs 2 and 3 of article 41, only authorizes, firstly, "copies and reproductions strictly reserved for use by copyists and not for general use" and, secondly, "analyses and short quotations for the purpose of example and illustration". Therefore, "any representation or reproduction, entire or partial, made without the consent of the author or his representatives is illegal" (paragraph 1 of article 40).

Any such representation or reproduction, made in any manner whatsoever, would therefore constitute an infringement of the law as sanctioned by articles 425 and in accordance with the penal code.

Information contained in this document is subject to change without prior notice and does not constitute any form of obligation on the part of OneAccess.

OneAccess and the distributors can in no case be held responsible for direct or indirect damage of any kind incurred as a result of any error in the software or guide.

Every care has been taken to ensure the exactitude of information in this manual. If however you discover an error, please contact OneAccess After Sales Service division.

Tenth edition: July 2011

1 INTRODUCTION

This edition of the OneOS Book corresponds to the V4.3R4 software release of OneOS. It is also available for all previous software releases of OneOS according to the features matrix described hereafter.

The OneOS software developed for use with the ONE product range offers an extensive range of features designed to provide a complete & highly powerful range of multi-service access routers:

- Full IP router with NAPT, Security, and Quality of Service management
- Support of voice for analog and ISDN S0/T0 terminals using Voice over IP and Voice over ATM
- Interworking of data protocols (FR, X.25, PAD, XOT, X.31)
- Advanced management tools based on CLI (Command Line Interface), SNMP, FTP/TFTP

This document is the OneOS user guide for WAN functions of the OneOS-based range products.

Seven other user guides and two global indexes are available:

- OneOS – Admin User Guide
- OneOS – Bridging & LAN User Guide
- OneOS – Basic IP User Guide
- OneOS – Advanced IP User Guide
- OneOS – VoIP User Guide
- OneOS – VoATM & CES User Guide
- OneOS – IBC User Guide
- Index of OneOS User Guides (global table of contents)
- Index of CLI of OneOS User Guides (global list of CLI commands)

1.1 FEATURE MATRIX

The following table is a resource providing edition by edition the released features. The table was done as of the V3.5R2E3 release. For simplification, the indicated software release shows the presence of a feature in a given software release starting with V3.5R2E3. It should be noted that most features were available in earlier versions.

Main Function	Feature	Present at least in:
WAN Interfaces	G.SHDSL 2 wires/4 wires	V3.5R2E3
	G.SHDSL 2/4 wires auto-sensing	V3.5R4E3
	G.SHDSL.bis for devices with ATM and EFM	V3.7R11E14
	SDSL 2B1Q (fixed rate)	V3.5R2E3
	STM-1	V3.5R2E3
	IMA E1	V3.5R2E3
	IMA SHDSL	V3.6R8E22 (ONE300)
	ADSL	V3.5R2E3
	Alternate ADSL FW loading	V4.2R5E2
	VDSL	V4.3R4E15
	ADSL/VDSL daughterboard	V4.3R4E15
	E1/T1 ATM	V3.5R2E3
	Serial E1/T1 (G.703 or G.704) and serial Vxx (PPP, MLPPP, FR)	V3.5R2E3
	Single ISDN S0 (ONE30/60/200): PPP, MLPPP	V3.5R2E3
	NxS0 on the ONE200 and ONE400: PPP, MLPPP	V3.5R2E3
	Single PRI on the ONE200 (fractional support) / ONE400 (full support): PPP, MLPPP	V3.5R2E3
	PRI framing auto detection	V4.2R4E2
	Analog PSTN modem (ONE30/60/200)	V3.5R2E3
	Analog PSTN modem (ONE100)	V3.6R5E3
	GPRS EDGE modem (ONECell25)	V3.6R10E18
	UMTS modem (ONECell35)	V3.7R13E6
	UMTS module firmware download	V4.2R5E15
ATM	IPoA: LLC or Mux encapsulation	V3.5R2E3
	PPPoA: unnumbered IP, PAP/CHAP authentication (optional 2-way authentication), encrypted password, LLC/Mux encapsulation	V3.5R2E3
	MLPPPoA: single PVC support, fragmentation and interleaving	V3.5R2E3
	PPPoEoA	V3.5R2E3
	Multi PPPoE on ATM-AAL5 + VLAN	V4.3R4E10
	ATM CoS: UBR, VBR, CBR	V3.5R2E3
	Multi VLAN in an ATM (generic) PVC	V4.2R2E2
ATM OAM	Automatic or manual F5 loopback	V3.5R2E3
	Configuration of continuity check cell	V3.5R2E3
	AIS/RDI management	V3.5R2E3
EFM	EFM configuration	V3.7R11E14
	Span monitoring	V5.1R2E2
	SNR, ES and SES thresholds monitoring	V5.1R2E2

Main Function	Feature	Present at least in:
	SNR, ES and SES statistics	V5.1R2E2
	SNR, ES and SES alarms traps	V5.1R2E2
PPP	Unnumbered IP	V3.5R2E3
	PAP/CHAP authentication, calling/called/two-authentication	V3.5R2E3
	PPP keep alive configuration	V3.5R2E3
	Scheduled reconnection	V3.5R2E3
	Random timer for PPP re-establishment	V3.5R2E3
	Password encryption	V3.5R2E3
	Static and dynamic IP	V3.5R2E3
	Link fragmentation and interleaving (LFI)	V3.5R2E3
	IPCP sub-net mask	V4.2R4E2
PPPoE	Multiplexed IPoE, PPPoE and IPoE	V3.5R2E3
	Multi PPPoE on EFM & Ethernet + VLAN	V4.3R4E10
Frame Relay	LMI: NUI, UNI interface types; Q.933 and ANSI mode	V3.5R2E3
	Frame relay traffic shaping: 4 priority levels, FRF.12 fragmentation and interleaving	V3.5R2E3
	One IP address per DLCI (no point-to-multipoint topology)	V3.5R2E3
ISDN interface	PPP and MLPPP (one single MLPPP bundle is allowed)	V3.5R2E3
	Dialing types: dial-in, dial-out, both	V3.5R2E3
	Dial-on-demand and permanent dial-out	V3.5R2E3
	ISDN signaling decoding	V3.5R2E3
	ISDN test calls	V3.5R2E3
	ISDN call-back	V3.5R2E3
	Authentication of the caller's number (no check or check a list of configured calling numbers)	V3.5R2E3
	Dial-out: main dialed number and backup ISDN numbers	V3.5R2E3
	BACP protocol (Bandwidth allocation Control Protocol)	V3.5R2E3
	Bandwidth on demand (open B-channels when needed)	V3.5R2E3
Dialer group	Activity monitoring of switched interfaces (to close the dial-out interface in case of inactivity)	V3.5R2E3
	Applicable to ISDN and PSTN interfaces	V3.5R2E3
Dialer watch-list	Open an interface when monitoring missing routes	V3.5R2E3
	Applicable to ISDN	V3.5R2E3
Router authentication	Via ISDN interface	V3.5R2E3
	Via PSTN	V3.5R2E3
Interworking functions	FRF.5 (Vxx serial interface of the ONE60/200)	V3.5R2E3
	FRF.8 (Vxx serial interface of the ONE60/200)	V3.5R2E3
	XOT (ONE30, Vxx serial interface of the ONE60/200, V.28 daughter board of the ONE60/200)	V3.5R2E3
	PAD over XOT (ONE30 V.28, V.28 daughter board of the ONE60/200)	V3.5R2E3
	X.25 over ISDN D channel (X.31)	V4.2R5E6
	X.25 call routing	V3.5R2E3
	X.25 address translation, support of CUG	V3.5R2E3
	X.25: configuration of source address	V3.5R2E3
	DNS-based XOT destination resolution	V3.5R2E3
	Serial COM port server	V4.3R4E11

1.2 TABLE OF CONTENTS

1 INTRODUCTION 1.1-3

1.1 Feature Matrix 1.1-4
1.2 Table of Contents 1.2-6

2 ATM BASED INTERFACES 1.2-11

2.1 ATM Interface Reference 2.1-11
2.2 ATM Interface Provisioning 2.2-11
2.3 ATM Based Interface Physical Configuration 2.3-13
  2.3.1 G.SHDSL for devices with ATM only 2.3-13
  2.3.2 Inverse Multiplexing for ATM (IMA) over G.SHDSL 2.3-17
    2.3.2.1 Principles of IMA 2.3-17
    2.3.2.2 Configuration 2.3-17
    2.3.2.3 Statistics 2.3-18
    2.3.2.4 Example 2.3-19
  2.3.3 G.SHDSL.bis for devices with ATM and EFM 2.3-20
    2.3.3.1 Setting up a SHDSL controller 2.3-20
    2.3.3.2 Setting up a DSL-group 2.3-20
    2.3.3.3 Applying the DSL group in the ATM or EFM interface 2.3-20
    2.3.3.4 Configuration command overview of the DSL group 2.3-21
    2.3.3.5 DSL group configuration commands 2.3-21
    2.3.3.6 DSL group monitoring configuration commands 2.3-24
    2.3.3.7 Additional DSL group configuration commands 2.3-25
    2.3.3.8 DSL group configuration default commands 2.3-26
    2.3.3.9 Statistics and Traces 2.3-27
      2.3.3.9.1 DSL group with ATM statistics 2.3-27
      2.3.3.9.2 DSL group with EFM statistics 2.3-28
      2.3.3.9.3 DSL group monitoring statistics 2.3-30
    2.3.3.10 G.SHDSL.bis configuration example 2.3-30
  2.3.4 Inverse Multiplexing for ATM (IMA) over G.SHDSL.bis 2.3-32
    2.3.4.1 Principles of IMA 2.3-32
    2.3.4.2 Configuration 2.3-32
    2.3.4.3 Statistics 2.3-32
    2.3.4.4 Example 2.3-32
  2.3.5 E1/T1 ATM Configuration 2.3-33
    2.3.5.1 Configuration Example 2.3-33
  2.3.6 Inverse Multiplexing for ATM (IMA) over E1/T1 .... 2.3-34
    2.3.6.1 Principles of IMA 2.3-34
    2.3.6.2 Configuration 2.3-34
    2.3.6.3 Statistics 2.3-36
    2.3.6.4 Example 2.3-36
  2.3.7 ADSL 2.3-37
    2.3.7.1 ADSL Configuration 2.3-37
    2.3.7.2 ADSL Statistics 2.3-38
    2.3.7.3 ADSL/VDSL daughterboard management 2.3-40
  2.3.8 SDSL 2B1Q Configuration 2.3-41
    2.3.8.1 Recommended Values Example for the Lucent Stinger DSLAM 2.3-42
  2.3.9 STM-1 .... 2.3-43
    2.3.9.1 Configuration 2.3-43
    2.3.9.2 Statistics 2.3-44
    2.3.9.3 Configuration Example 2.3-44

2.4 ATM Generic Permanent Connections (Generic PVC) 2.4-45
  2.4.1 PVC Templates 2.4-45
  2.4.2 Generic PVC Creation 2.4-46
  2.4.3 Generic PVC Configuration 2.4-46
  2.4.4 Generic PVC statistic counters reset 2.4-47
  2.4.5 Generic PVC configuration example 2.4-47

2.5 ATM Dedicated Permanent Connections (Dedicated PVC) 2.5-48
  2.5.1 Dedicated PVC Attachment 2.5-48
  2.5.2 Dedicated PVC Configuration 2.5-48
    2.5.2.1 IP over ATM Service 2.5-49
    2.5.2.2 PPP over ATM Encapsulation 2.5-49
      2.5.2.2.1 Configuration 2.5-50
      2.5.2.2.2 IP Address 2.5-50
      2.5.2.2.3 Encapsulation Type 2.5-50
      2.5.2.2.4 IPCP Parameters 2.5-50
      2.5.2.2.5 MRU Parameters 2.5-50
      2.5.2.2.6 Authentication Type 2.5-50
      2.5.2.2.7 Username and Password 2.5-51
      2.5.2.2.8 Magic Number Negotiation 2.5-51
      2.5.2.2.9 Keep-Alive Timer 2.5-51
      2.5.2.2.10 Retry Timer 2.5-51
      2.5.2.2.11 Link Fragmentation and Interleaving (LFI) 2.5-52
      2.5.2.2.12 PPPoA session re-initialization 2.5-52
      2.5.2.2.13 Example 2.5-53
      2.5.2.2.14 Statistics 2.5-53
    2.5.2.3 PPP over Ethernet over ATM Encapsulation (PPPoEoA) 2.5-53
      2.5.2.3.1 Configuration 2.5-54
      2.5.2.3.2 Example 2.5-54
      2.5.2.3.3 Statistics 2.5-55
  2.5.3 Multi-Link PPP (MLPPP) over ATM and MLPPPoEoA 2.5-56
    2.5.3.1 Configuration 2.5-56
    2.5.3.2 Example 2.5-57
    2.5.3.3 Debug and Statistics 2.5-59

2.6 ATM Management Flows 2.6-60
  2.6.1 Features 2.6-60
    2.6.1.1 Definitions 2.6-60
      2.6.1.1.1 OAM 2.6-60
      2.6.1.1.2 OAM Flow 2.6-60
      2.6.1.1.3 Description of OAM Information Flow 2.6-60
      2.6.1.1.4 OAM Cell Structure 2.6-60
    2.6.1.2 Handling of ATM OAM 2.6-61
      2.6.1.2.1 ATM OAM Prioritization 2.6-61
      2.6.1.2.2 Continuity Check (CC) 2.6-61
      2.6.1.2.3 Loopback (LB) 2.6-61
      2.6.1.2.4 Alarm Indication Signal (AIS) and Remote Defect Information (RDI) 2.6-62
  2.6.2 Permanent Virtual Channel Connectivity 2.6-62
    2.6.2.1 Loopback Ping 2.6-62
    2.6.2.2 Permanent Virtual Channel Configuration (PVC/SVC) 2.6-62
    2.6.2.3 Management of a Virtual Channel 2.6-62
    2.6.2.4 OAM F5 Loopback End-to-End 2.6-62
    2.6.2.5 F5 OAM Loopback Segment 2.6-63
    2.6.2.6 F5 OAM Continuity Check End-to-End 2.6-63
    2.6.2.7 F5 OAM Continuity Check Segment 2.6-63
    2.6.2.8 F5 OAM AIS RDI 2.6-63
    2.6.2.9 F5 OAM Intrusive Management 2.6-64
    2.6.2.10 Configuration Example 2.6-64
    2.6.2.11 Statistics 2.6-65

3 ETHERNET IN THE FIRST MILE – EFM 2.6-66

3.1 Introducing EFM 3.1-66

3.2 EFM OAM – Operations, Administration and Maintenance 3.2-67
  3.2.1 Purpose of OAM 3.2-67
  3.2.2 OAM PDUs or protocol data units 3.2-67
  3.2.3 OAM discovery 3.2-67
  3.2.4 OAM remote loopback 3.2-67
  3.2.5 OAM Active mode 3.2-68
  3.2.6 OAM Passive mode 3.2-68

3.3 EFM Configuration 3.3-69
  3.3.1 EFM configuration commands for G.SHDSL.bis links 3.3-69
  3.3.2 EFM configuration commands for VDSL2 links 3.3-70
  3.3.3 EFM configuration commands as an Ethernet interface 3.3-70
  3.3.4 VDSL 3.3-71

3.3.4.1 VDSL Configuration....3.3-71
3.3.4.2 VDSL configuration example....3.3-72
3.3.4.3 VDSL statistics....3.3-72
3.3.4.4 ADSL/VDSL daughterboard management....3.3-75
3.4 EFM Debug and Statistics....3.4-76

4 SERIAL HDLC INTERFACES....3.4-77

4.1 V.11/V.35/V.28/V.36 Serial Interfaces....4.1-77
4.2 E1/T1 Interfaces....4.2-79
4.3 PPP over Serial Line Configuration....4.3-80
4.3.1 Debugging PPP over Serial....4.3-81
4.4 Frame Relay over Serial Interface....4.4-81
4.4.1 Features....4.4-81
4.4.2 Configuration....4.4-81
4.4.2.1 LMI Configuration....4.4-82
4.4.2.2 Completing Serial Interface Configuration....4.4-82
4.4.2.3 Frame-Relay QoS Configuration (Optional)....4.4-82
4.4.2.4 DLCI Configuration (Required)....4.4-83
4.4.2.5 PVC Prioritization....4.4-83
4.4.3 Statistics....4.4-85
4.5 Serial Line Statistics....4.5-86

5 PPP VIRTUAL TEMPLATE CONFIGURATION....4.5-87

5.1 Bi-Directional CHAP & CHAP Server configuration....5.1-87
5.2 PPP Client Configuration....5.2-87
5.2.1 Entering in PPP Virtual-Template Configuration Mode....5.2-87
5.2.2 IPCP Parameters....5.2-87
5.2.3 IP Address....5.2-88
5.2.4 Authentication Type....5.2-88
5.2.5 Username and Password....5.2-88
5.2.6 Link Fragmentation and Interleaving (LFI)....5.2-89
5.2.7 Completing Template configuration....5.2-89
5.2.8 Examples....5.2-89
5.3 PPP Server Configuration....5.3-91
5.3.1 IPCP Parameters....5.3-91
5.3.2 IP Address....5.3-91
5.3.3 IP Address Pool....5.3-91
5.3.4 RADIUS Server....5.3-91
5.3.5 Completing Template configuration....5.3-92
5.4 Common Parameters for PPP Client and PPP Server....5.4-92
5.4.1 MRU Parameters....5.4-92
5.4.2 Magic Number Negotiation....5.4-92
5.4.3 Keep alive Timer....5.4-93
5.4.4 Reconnection....5.4-93
5.4.5 Retry Timer....5.4-93

6 PPPOE ON ETHERNET UPLINK....5.4-94

6.1 PPPoE Configuration....6.1-94
6.2 PPPoE Configuration Example....6.2-97
6.3 PPPoE Statistics and Traces....6.3-97

7 ISDN CONFIGURATION FOR DATA SERVICES....6.3-98

7.1 ISDN Data Service Description....7.1-98
7.2 ISDN Physical Layer Configuration....7.2-100
7.2.1 ISDN Signaling Configuration....7.2-100
7.2.2 PRI Interface Configuration....7.2-101
7.2.3 B-Channel Configuration....7.2-102
7.3 ISDN Service Configuration....7.3-103
7.3.1 Dial-out Calls Configuration....7.3-103
7.3.2 Dial-in Calls Configuration (OneOS-based router being called)....7.3-103
7.3.3 Bi-Directional Calls Configuration....7.3-104
7.3.4 Common Calls parameters Configuration....7.3-104
7.4 ISDN Statistics....7.4-106

7.5 ISDN Debug and Traces 7.5-116
7.6 ISDN Configuration Examples 7.6-116

8 CONFIGURATION OF PSTN MODEM 7.6-119

8.1 PSTN Modem Service Description 8.1-119
8.2 PSTN Modem Configuration 8.2-119
8.2.1 Physical Interface Parameters 8.2-119
8.2.2 Interface Backup Routing 8.2-120
8.2.3 Multipath Backup Routing 8.2-120
8.2.4 Common Link Parameters 8.2-120
8.3 PSTN Modem Statistics and Traces 8.3-121
8.4 PSTN Modem Configuration Example 8.4-122

9 INTERNAL EDGE/GPRS MODEM 8.4-123

9.1 EDGE/GPRS Modem PIN Code 9.1-123
9.2 EDGE/GPRS ConnectioN Configuration 9.2-124
9.3 EDGE/GPRS Configuration Examples 9.3-125
9.4 EDGE/GPRS Statistics and Troubleshooting 9.4-126

10 INTERNAL UMTS MODEM 9.4-127

10.1 UMTS Modem PIN Code 10.1-127
10.2 UMTS Connection Configuration 10.2-128
10.3 UMTS Modem Configuration Examples 10.3-129
10.4 UMTS Modem Statistics and Troubleshooting 10.4-131
10.5 UMTS module firmware download 10.5-131

11 EXTERNAL MODEM USING AT COMMANDS (GPRS) 10.5-132

12 AUTHENTICATION OF DSL LINK BASED ON PSTN/ISDN 10.5-133

12.1 PSTN/ISDN Authentication Configuration 12.1-133
12.2 PSTN/ISDN Authentication Example 12.2-134
12.3 PSTN/ISDN Authentication Traces and Debug 12.3-134

13 ACTIVITY CONTROL OF BACKUP & SWITCHED INTERFACES 12.3-136

13.1 Introduction to Switched Interface 13.1-136
13.2 Dialer-Groups 13.2-136
13.3 Backup using Route Monitoring (Dialer Watch-List) 13.3-139

14 INTERWORKING FUNCTIONS FOR LEGACY PROTOCOLS 13.3-143

14.1 FR - ATM Interworking 14.1-143
14.1.1 FRF.5 Frame Relay-ATM Network Interworking 14.1-143
14.1.1.1 Principle 14.1-143
14.1.1.2 FRF.5 Configuration 14.1-144
14.1.1.3 One to one Interworking Connection 14.1-144
14.1.1.3.1 Configuration of the ATM interface 14.1-144
14.1.1.3.2 Connection Frame Relay to ATM 14.1-144
14.1.1.4 Many to one Interworking Connection 14.1-145
14.1.1.4.1 Configuration of the ATM interface 14.1-145
14.1.1.4.2 Connection Frame Relay to ATM 14.1-145
14.1.1.5 Configuration example 14.1-146
14.1.2 FRF.8 Interworking 14.1-146
14.1.2.1 Principle 14.1-146
14.1.2.2 FRF.8 Configuration 14.1-147
14.1.2.3 Serial Interface Configuration 14.1-147
14.1.2.4 Configuration of the ATM interface 14.1-147
14.1.2.5 Connection Frame Relay to ATM 14.1-147
14.1.2.6 Configuration example 14.1-148
14.1.2.7 Status and Statistics Commands 14.1-149
14.2 X.25 over TCP/IP (XOT) 14.2-152
14.2.1 Configuration Commands 14.2-153
14.2.1.1 Serial Interface 14.2-153
14.2.1.2 X.25 Layer-2 .... 14.2-153

14.2.1.3 X.25 Layer-3 Configuration....14.2-154
14.2.1.4 X.25 Routing....14.2-158
14.2.1.5 XOT Routing Table....14.2-158
14.2.1.6 Statistics and Configuration Display....14.2-161
14.2.1.7 Debug and Traces....14.2-163
14.2.1.8 Configuration Example....14.2-163
14.2.2 PAD over XOT....14.2-164
14.2.2.1 X.3 Parameter Configuration....14.2-165
14.2.2.1.1 Introduction....14.2-165
14.2.2.1.2 Configuration....14.2-166
14.2.2.1.3 Example....14.2-167
14.2.2.2 V.28 Interface Configuration....14.2-167
14.2.2.2.2 PAD <---> X.25 Layer Configuration....14.2-168
14.2.2.2.3 Statistics....14.2-169
14.2.2.2.4 Example....14.2-169
14.3 X.25 over ISDN (X.31)....14.3-170
14.3.1 Configuration commands....14.3-170
14.3.1.1 ISDN D channel interface....14.3-170
14.3.1.2 X.25 layer 3 template....14.3-171
14.3.1.3 X.25 D channel interface (X.31 interface)....14.3-171
14.3.1.4 X.31 interface routing....14.3-172
14.3.1.5 Statistics and Configuration Display....14.3-172
14.3.1.6 Debug and Traces....14.3-174
14.3.1.7 Configuration Example....14.3-174
14.4 Serial Com Port Server....14.4-176
14.4.1 Serial COM Port Server License....14.4-176
14.4.2 Serial cable handling....14.4-176
14.4.2.1 At start-up....14.4-176
14.4.2.2 When the device is running....14.4-176
14.4.2.2.1 On "serial" cable detection....14.4-176
14.4.2.2.2 On "serial" cable disconnection....14.4-177
14.4.3 TCP connection....14.4-177
14.4.4 Software architecture....14.4-177
14.4.5 COM Server configuration....14.4-177
14.4.6 COM Server Configuration Display....14.4-180
14.4.7 COM Server status display....14.4-180
14.4.8 Statistics Display....14.4-180
14.4.9 Debug and Traces....14.4-181
14.4.10 Configuration Example....14.4-182

2 ATM BASED INTERFACES

The ATM service enables the transport of voice and data. To initiate the service, protocol stacks are configured using CLI commands. These commands permit the configuration of the ATM physical interface and logical sub-interfaces that are associated with Permanent Virtual Connections (PVC).

The PVC configuration requires parameters related to ATM technology (VPI, VCI, QoS, OAM, etc.) and parameters related to the type of transport protocol (classical IP or IP over PPP).

The steps for configuration are:

- Parameter settings (value control)
- Checking parameter coherence and applying changes (command: execute).

If the** execute **command is not entered, WAN parameters do not enter in effect.

2.1 ATM INTERFACE REFERENCE

The ATM interface is referenced by the port number as explained in the previous section.

The port number references the interface that gathers the physical interface (such as DSL) and several PVC supported by this interface.

Enter the port number to configure the ATM interface (the interface is created if it does not exist):

```
CLI> configure terminal
CLI(configure)> interface atm <port number>
CLI(config-if)>
```

2.2 ATM INTERFACE PROVISIONING

The following commands set common PVC parameters, such as the chosen VP/VC identifiers.

The parameters are assembled in a profile defined by “`driver ident`”. Enter the `driver ident` index (usually 0):

```
CLI(config-if)> driver ident <index>
CLI(drv)>
```

Then specify the maximum number of PVC (VP/VC) that can be provisioned on the interface:

```
CLI(drv)> max channels <value>
```

`value` parameter indicates the maximum PVC number available under the interface. Default: 8.

Then, enter the maximum VP number and the maximum VC number:

```
CLI(drv)> max vp <value>
CLI(drv)> max vc <value>
```

The first command provides the maximum number of Virtual Paths (VP) allowed on this interface. The number varies between 1 and 8. Default: 1.

The second command provides the maximum number of Virtual Circuits (VC) allowed. The number varies between 1 and 8. Default: 1.

The range of value for VP and VC identifiers must be set using the following commands:

```
CLI(drv)> range vp min <value> max <value>
```

The above-referenced command specifies the minimal and maximal bounds of the VP Identifier (VPI) value allowed. This VPI can vary between 0 and 255. Default min: 0, Default max: 1.

```
CLI(drv)> range vc min <value> max <value>
```

The above-referenced command specifies the minimal and maximal bounds of the VC Identifier (VCI) value allowed in configuration. The VCI can vary between 32 and 2048. Default min: 32, Default max: 40.

The Maximum Transmit Unit (MTU) size can be specified. The MTU designates the largest packet size that is supported by the interface. The MTU settings apply to data traffic using the interface. The command is:

```
CLI(drv)> mtu <size>
```

The size ranges between 32 and 9188 (size in bytes). Default: 4470.

Example:

```
interface atm 0
driver ident 0
max vp 1
max vc 3
range vc min 32 max 100
range vp min 0 max 5
max channels 3
mtu size 1496
execute
exit
exit
```

Physical layer configuration of ATM based interface is described in the following sections according to physical layer type.

When configured, use the following command in global mode to reset the physical layer (e.g. DSL layer):

```
CLI> clear synchronize interface atm 0
```

2.3 ATM BASED INTERFACE PHYSICAL CONFIGURATION

2.3.1 G.SHDSL for devices with ATM only

The G.SHDSL interface corresponds to the most popular form of symmetrical DSL. It is based on the G.991.2 standard and offers better reach than the SDSL 2B1Q standard.

To start the G.SHDSL configuration, the user must enter into the configuration mode using the command:

```
CLI> configure terminal
```

Then the user must enter into the interface configuration mode using the command:

```
CLI(configure)> interface atm <interface number>
```

The CLI enters into the G.SHDSL configuration mode (in `config-if`), when typing the next command:

```
CLI(config-if)> gshdsl
```

This command removes the interface G.SHDSL:

```
CLI(config-if)> no gshdsl
```

Some parameters must be chosen before starting the G.SHDSL interface. The related commands and parameters are:

The annex command specifies the standard that will be used: A, for American standard and B, for European standard.

```
CLI(gshdsl)> annex { A | B }
```

The following command indicates the equipment type. Only the CPE type is offered.

```
CLI(gshdsl)> equipment CPE
```

The following command enables configuration of the interface bit rate in fixed mode. The rate value is fixed between 192 and 2304 kbps by steps of 64 kbps (2-wire G.SHDSL) or between 384 and 4608 kbps by steps of 128 kbps (4-wire G.SHDSL).

```
CLI(gshdsl)> linerate fixed <value>
```

The following command sets the range of bit rates that the G.SHDSL chipset can use in adaptive mode. The device and DSLAM provide the highest line rate possible within the range defined here according to line features (length, attenuation) and tolerable Bit Error Ratio. Two parameters define the minimum rate and the maximum rate (in kbps).

```
CLI(gshdsl)> linerate adaptive <value1> <value2>
```

To use the default bit rate setting (adaptive mode from 192 to 2304 kbps in 2-wire mode and from 384 to 4608 kbps in 4-wire mode):

```
CLI(gshdsl)> default linerate
```

A G.SHDSL interface can use two or four wires; doubling the number of wire doubles the interface speed. The following command forces a four-wire interface to either apply the two-wire mode or to specify the four-wire operation mode. Only the four-wire interface with byte interleaving is supported (bit interleaving is not supported). When using the interface in four-wire mode, three connection modes are possible: `standard`, `enhanced` or `enhanced_sim`. The `enhanced` mode synchronizes one copper pair then the other. The `enhanced_sim` and the `standard` mode start the synchronization of both pairs simultaneously. These two modes enable then a faster synchronization time. The standard mode is compliant with the ITU-T standard. Therefore, the standard mode is the default value. In case of connection problems, it is recommended to try all of them.

```
CLI(gshdsl)> mode { 2_wire | 4_wire_byte_interleave [enhanced
                                                      | enhanced_sim | standard] }
```

When the G.SHDSL interface has four wires, you can configure the `autoconfig` mode. In that mode, the router detects the DSLAM configuration and then sets the appropriate G.SHDSL parameters (unless some parameters are forced in the CLI configuration). The argument is the connection mode used for the handshake. The handshake can be done in two or four-wire mode, then the CPE retrieves the line parameters from the DSLAM (line rate, number of wires, enhanced or standard, first pair to synchronize) and initializes the G.SHDSL line according to the retrieved parameters. If you previously forced a parameter like the number of wires (command: `mode 2_wire`) or line rate (command: `linerate`), the retrieved corresponding parameter from the DSLAM is ignored.

```
CLI(gshdsl)> mode autoconfig { 2-wire | 4-wire-enhanced
                                      | 4-wire-standard }
```

A `mode autoconfig synchro-GSPN` command has been added for inter vendor compatibility. For detailed information about this command, contact OneAccess Support.

To disable `autoconfig`:

```
CLI(gshdsl)> no mode autoconfig
```

The mode autosense has the same effect as `mode autoconfig` but it is a deprecated command.

```
CLI(gshdsl)> mode autosense
```

The `mode-dslam` command is actually a workaround to allow a fast adaptive connection with DSLAM using a `Metallink` chipset (default setting: default).

```
CLI(gshdsl)> mode-dslam { default | METALLINK }
```

The following command closes the physical interface configuration applying the parameters that have been set.

```
CLI(gshdsl)> execute
```

Example:

```
interface atm 0
 gshdsl
 execute
 exit
exit
```

Other commands: only for experts

The `max-tx-ring` command specifies the maximum size of the transmission buffer ring. The minimum configurable size is 4 and the maximum is 128. The default size is 128.

```
CLI(gshdsl)> max-tx-ring <value>
```

The `action-sync-lock` command is a workaround command and specifies the action to do in case of deadlock during synchronization: `none`, `reboot` or `resync` for beginning a new synchronization can be set as parameter.

```
CLI(gshdsl)> action-sync-lock { none | reboot | resync }
```

The `sync-pair` command forces the first pair that will be started at beginning of synchronization. No pair is forced while setting `auto`, first pair with `normal` and second pair with `crossed`. Default value is `auto`.

```
CLI(gshdsl)> sync-pair { auto | normal | crossed }
```

The `synchro` command specifies the activation of retrain on loss of framing (if framing is lost, by default, the SHDSL line should restart training). Retrain is enabled if setting `synchro` is set as `check` otherwise it is disabled.

```
CLI(gshdsl)> synchro { check | default | no }
```

By default, the number of CRC errors is monitored. By default, the lines are resynchronized if the SHDSL modem detects more than 3 CRC errors per second. Entering "`crc check <n>`" changes the number of acceptable consecutive CRC errors during one second. Its value varies from 1 to 30. "`crc check`" sets the default CRC error value. "`crc no`" disables the control of CRC errors (resynchronization is started due to other criteria such as loss of signal).

```
CLI(gshdsl)> crc { check [<n>] | no }
```

Example:

```
interface atm 0
 driver ident 0
  range vp min 0 max 100
  range vc min 32 max 100
  execute
 exit
 gshdsl
  crc check 30
  execute
 exit
exit
```

To activate traces, use the following command:

```
CLI> trace filter add sys hdsl all
```

The SHDSL statistics are viewed as follows:

```
CLI> show gshdsl
```

The SHDSL statistics are interpreted as follows:

<table>
<tr><th>show gshdsl</th><th>Interpretation</th></tr>
<tr><td><pre>
Admin.Status UP   Oper.Status UP UP
</pre></td><td></td></tr>
<tr><td><pre>
Line is configured by user:
  ifIndex = 0
  Mode is CPE (Annex B)
  Configured rate is ADAPTIVE from 384 K to 4608 K
  Noise margin is 2 dB
  Snext noise margin is Disabled
  Autoconfig 4-wire-enhanced
  Frame sync not checked
</pre></td><td>Items configured via CLI</td></tr>
<tr><td><pre>
Phase: line UP
Operational Mode: 4-wire mode enhanced
Remote provider code: NPSG (4e 50 53 47)
Remote firmware version: 0x20
Remote configured rate is FIXED at 2048 K
</pre></td><td>Synchronization phase: IDLE, autoconf (detecting the line parameters), synchro (synchronization in progress), UP
DSLAM parameters</td></tr>
<tr><td><pre>
First pair started is 0
pair[0] [DATA   ][NO_STARTUP]
pair[1] [DATA   ][NO_STARTUP]
</pre></td><td>Shows the ID of the first line being synchronized (when they must not be synced simultaneously).</td></tr>
<tr><td><pre>
Retrain criteria:
              (value   ref   enable)
pair[0]   SQ 38.76 22.70   ON
          CRC   0   3   ON
        FSync   0   0   OFF
pair[1]   SQ 38.19 22.70   ON
          CRC   1   3   ON
        FSync   1   0   OFF

Retrain counters:
           (SQ   CRC FSync)
pair[0]   0   0   0
pair[1]   0   0   0
Line   0   0   0
</pre></td><td>Line Monitoring Statistics:
They are the criteria that can force the interface to restart synchronization and training. By default the SQ (Signal Quality) must be greater than the reference value. CRC must remain low and framing Synchronization (Fsync) must be active. The CRC counter is the number of CRC errors within a second. The ‘enable’ column indicate if the corresponding criterion (SQ, CRC, Fsync) can force the interface to retrain SHDSL.
The second part shows the number of resynchronization caused by bad signal quality (SQ), excessive CRC errors or loss of framing sync. The counters are shown for every pair and the whole line (combining both pairs).</td></tr>
</table>

show gshdsl	Interpretation
Mib status 0x00000001 ,noDefect Mib status 0x00000001 ,noDefect Data Rate: 1024000 bps 1024000 bps Data Rate Ratio: 266% Attenuation: 0,0 dB 0,1 dB NoiseMargin: 38,4 dB 38,2 dB TransmitPwr: 7,5 dB 7,5 dB ReceiverGain: 5,8 dB 5,7 dB chipset release: 0 0 firmware release: [R3.0.5]	Line statistics SHDSL firmware release
(power-on lastread) CRC 0x00000000 0x00000000 RxCells 0x0000001e 0x0000001e RxDrops 0x00000000 0x00000000 RxHec 0x00000000 0x00000000 TxCells 0x0000001e 0x0000001e	ATM cell counters CRC errors Received ATM cells Dropped received ATM cells Received ATM cells with HEC error Number of sent ATM cells
(power-on lastread) ES 0x00000000 0x00000000 SES 0x00000000 0x00000000 LOSWS 0x00000000 0x00000000 UAS 0x00000001 0x00000001	ITU Standard Counters (see G991.2) ES: Number of Errored Seconds (second with at least one error) SES: Severely Erred Seconds (number of seconds with high CRC rate) LOSWS: number of seconds with loss of synchronization word UAS: unavailable seconds (number of seconds with line outage)
EOC is enabled: expecting discovery EOC is enabled: expecting discovery EOC statistics (power-on lastread) (power-… commands 0x00000000 0x00000000 0x00000… responses 0x00000000 0x00000000 0x00000… discarded commands 0x00000000 0x00000000 0x00000… errored commands 0x00000000 0x00000000 0x00000… received frames 0x00000000 0x00000000 0x00000… bad transparency 0x00000000 0x00000000 0x00000… bad FCS 0x00000000 0x00000000 0x00000…	EOC Statistics Stable state is: “EOC is enabled: running”

2.3.2 Inverse Multiplexing for ATM (IMA) over G.SHDSL

2.3.2.1 Principles of IMA

IMA is a technique enabling to split and reassemble an ATM cell stream over multiple physical links. It was defined by the ATM Forum recommendation AF-PHY-0086.0001. For more details, please refer to:

`http://broadband-forum.org/ftp/pub/approved-specs/af-phy-0086.001.pdf`

This technique is highly efficient to increase incrementally the capacity of transmission links. The ONE300/400 can be equipped with an IMA interface supporting four G.SHDSL links which allows a symmetric data rate up to 9139 Kbps.

An IMA interface forms an IMA group. An IMA group is actually made up of several physical links. The role of IMA is to split the incoming cell traffic over the different physical interfaces. The IMA group must respectively reassemble the cell stream at the remote end. The IMA algorithm ensures the cell stream is reassembled in the proper order and compensates for possible inter-link delays. An embedded OAM ensures links can be dynamically added / removed in the group.

The data transiting over the DSL links are made up of:

- ATM cells (the cells are sent over each link on a cell-by-cell basis)
- ICP (IMA Control Protocol) cells
- Filler cells (when no ATM cell have to be sent)

The IMA group manages the cell stream over the N links. Over each link, the IMA group controls the cell stream by using an IMA frame of length M. An IMA frame contains one ICP cell, some ATM cells and/or filler cells. The IMA frame is the unit of control of IMA.

2.3.2.2 Configuration

To provision the IMA interface, first enter in ATM interface configuration mode:

```
CLI(configure)> interface atm <interface-number>
CLI(config-if)> [no] ima
CLI(ima)>
```

`interface-number` is always 0.

The CLI has entered into the IMA configuration mode. You can set the IMA frame length M, when typing the following command:

```
CLI(ima)> frame-length { 32 | 64 | 128 | 256 }
```

The frame length must be the same at each end of the IMA link (default: 32 cells per frame). The following command sets the number of active links required to consider the interface as up and running:

```
CLI(ima)> active-link-number <1..4/8>
```

Default: `1`. Maximum value depends on the hardware configuration (one300 <4>, one400 <4/8>)

Inter-link differential delays are the difference between transit delays of each link. The IMA group must compensate such differential delays till a maximum configured as follows:

```
CLI(ima)> max-delay <milliseconds>
```

The default value is 20 ms:

The following command sets the clock property for links of the IMA group:

```
CLI(ima)> clock-mode { dependent | independent }
```

If `dependent` is selected, it means that all links are synchronized on the same clock source. `independent` indicates that links may have different source. Usually, all links are always synchronized on the same source; the default value is then `dependent`.

Then you have to set the common physical layer parameters (G.SHDSL). The following command makes the CLI enter in the common G.SHDSL configuration node:

```
CLI(ima)> gshdsl
CLI(shdsl)>
```

The following command configures the G.SHDSL annex type:

```
CLI(shdsl)> annex { A | B }
```

Default: "B". "A" is used in North America and "B" in Europe.

The following command indicates the equipment type. Only the CPE type is offered.

```
CLI(shdsl)> equipment CPE
```

The following command enables configuration of the interface bit rate in fixed mode or dynamic mode. The rate value is fixed between 192 and 2304 kbps by steps of 64 kbps or dynamic (get from remote DSLAM line rate).

```
CLI(shdsl)> linerate { <value> | dynamic }
```

Default is `linerate dynamic`.

Then enter `exit` to finish the common G.SHDSL parameters configuration.

Last you must configure each G.SHDSL link. The following command makes the CLI enter in G.SHDSL link interface configuration mode:

```
CLI(ima)> [no] gshdsl-link { <0..3> | <0..7> }
CLI(shdsl-link)>
```

Maximum value depends on the hardware configuration (one300 <0..3>, one400 <0..3/7>).

The following command specifies the activation of retrain on loss of framing (if framing is lost, by default, the SHDSL line should restart training). Retrain is enabled if setting `synchro` is set as `check` otherwise it is disabled.

```
CLI(shdsl-link)> synchro { check | no }
```

Then enter `exit` to finish the link configuration and configure the following links. Then, enter `execute`.

Other commands: only for experts

The `max-tx-ring` command specifies the maximum size of the transmission buffer ring. The minimum configurable size is 4 and the maximum is 128. The default size is 128.

```
CLI(ima)> max-tx-ring <value>
```

2.3.2.3 Statistics

IMA statistics are obtained using the following command (explanations about the meaning of counters is provided hereafter):

```
CLI> show ima
```

```
+-------------------------------------------------------------------------------------+
| UP Time duration: 0d 0h 8m 44s             Current time: 0d 1h 48m 28s              |
| Upstream: 4315472 b/s (10178 cell/s)       Downstream: 4315472 b/s (10178 cell/s)   |
+-------------------------------------------------------------------------------------+
| Group 1 UP            Ne State: Operational          Fe State: Operational          |
| Up/Down counter:   4  4/4 links active to transmit   4/4 links active to receive    |
+-------------------------------------------------------------------------------------+
| Link 0 UP      Tx Id : 0                          Rx Id : 2                         |
|*(RefClock TX)    NeTxState  Active                  NeRxState Active                |
|*(RefClock RX)    FeRxState  Active                  FeTxState Active                |
+-------------------------------------------------------------------------------------+
| Link 1 UP      Tx Id : 1                          Rx Id : 1                         |
|                  NeTxState  Active                  NeRxState Active                |
|                  FeRxState  Active                  FeTxState Active                |
+-------------------------------------------------------------------------------------+
| Link 2 UP      Tx Id : 2                          Rx Id : 0                         |
|                  NeTxState  Active                  NeRxState Active                |
|                  FeRxState  Active                  FeTxState Active                |
+-------------------------------------------------------------------------------------+
| Link 3 UP      Tx Id : 3                          Rx Id : 3                         |
|                  NeTxState  Active                  NeRxState Active                |
|                  FeRxState  Active                  FeTxState Active                |
+-------------------------------------------------------------------------------------+
```

The first block gives the following IMA group connection information:

- `Up time duration`: Elapsed time since IMA group goes UP.
- `Current time`: Elapsed time since last reboot.
- `Up/Downstream`: IMA Rx/Tx data rate in bps and IMA data cell rate in cell/s.

The second block gives the following IMA group status information:

- `Group <grpID>`: IMA group main state **{** `UP` **|** `DOWN` **}**.
- `Ne/Fe State`: Near-end/far-end IMA group state machine **{** `Start-up` **|** `Start-up Ack` **|** `Config aborted` **|** `Blocked` **|** `Insufficient Links` **|** `Operational` **}**.
- `Up/Down counter`: IMA group up/down counter since group creation.
- `Rw/Tx link active`: Quantity of links active to transmit or receive.

The next blocks give the following IMA links status information:

- `Link <lnkID>`: IMA link main state (G.SHDSL link state) **{** `UP` **|** `DOWN` **}**.
- `Top star '*'`: The physical clock source for the Transmit IMA group.
- `Bottom star '*'`: The physical clock source for the Receive IMA group.
- `RefClock Tx`: The line ID of the link to be used as the transmit timing reference.
- `RefClock Rx`: The Line ID of the link being used as the receive IMA group timing reference.
- `Tx/Rx Id`: Rx ID (FE link ID) / Tx ID (NE link ID).
- `Ne/FeTxState`: Near-end / far-end IMA transmit link state machine **{** `Not In Group` **|** `Unusable No Reason` **|** `Unusable Fault` **|** `Unusable Misconn` **|** `Unusable Blocked` **|** `Unusable Failed` **|** `Usable` **|** `Active` **}**.
- `Ne/FeRxState`: Near-end / far-end IMA receive link state machine **{** `Not In Group` **|** `Unusable No Reason` **|** `Unusable Fault` **|** `Unusable Misconn` **|** `Unusable Blocked` **|** `Unusable Failed` **|** `Usable` **|** `Active` **}**.

2.3.2.4 Example

The following example is an IMA interface with 4 G.SHDSL link at 4 x 2048 Kbps:

```
interface atm 0
 ima
  gshdsl
   linerate 2048
   exit
  gshdsl-link 0
   exit
  gshdsl-link 1
   exit
  gshdsl-link 2
   exit
  gshdsl-link 3
   exit
  execute
 exit
exit
```

2.3.3 G.SHDSL.bis for devices with ATM and EFM

The G.SHDSL.bis interface is based on the G.991.2 Annex G standard.

For devices with both ATM and EFM, configuring G.SHDSL.bis requires three steps. The three steps are further explained in the sections below:

1. Setting up a SHDSL controller
2. Setting up a DSL-group
3. Linking the DSL group to the ATM or EFM interface

2.3.3.1 Setting up a SHDSL controller

To start the G.SHDSL.bis configuration, the user must enter into the configuration mode using the command:

```
CLI> configure terminal
CLI(configure)>
```

The `controller` command enters the configuration mode of SHDSL controller:

```
CLI(configure)> controller shdsl 0
CLI(conf-ctrl)>
```

The `controller shdsl 0` command identifies the SHDSL interface.

2.3.3.2 Setting up a DSL-group

After the SHDSL controller has been set up, the next step is to set up a DSL group within the SHDSL controller.

The `dsl-group` command enters the configuration mode of the DSL group:

```
CLI(conf-ctrl)> dsl-group <dsl-group-number>
```

Currently, a single DSL group 0 can be configured.

We proceed with DSL group 0:

```
CLI(conf-ctrl)> dsl-group 0
CLI(dsl-group)>
```

Now, the DSL group can be configured. Refer to 2.3.3.4 Configuration command overview of the DSL group for a detailed explanation of the DSL group commands.

2.3.3.3 Applying the DSL group in the ATM or EFM interface

For an ATM interface, proceed as follows:

The following command enters the configuration mode of an ATM interface, for example 0:

```
CLI(configure)> interface atm 0
```

The following command assigns the configured DSL group to the ATM interface:

```
CLI(config-if)> dsl-group 0
CLI(config-if)>
```

For an EFM interface, proceed as follows:

The following command enters the configuration mode of an EFM interface, for example 0:

```
CLI(configure)> interface efm 0
```

Note: As of V4.2 software release, only one EFM interface, the EFM interface 0, can be activated.

The following command assigns the configured DSL group to the EFM interface:

```
CLI(conf-if)> dsl-group 0
CLI(conf-if)>
```

Also refer to 2.3.3.10 G.SHDSL.bis configuration example for a configuration example.

2.3.3.4 Configuration command overview of the DSL group

The following gives an overview of the DSL group configuration commands:

"Standard" configuration commands:

```
dsl-group
|
+---annex
+---autoconfig
+---caplist
+---cctargetmargin
+---crc
+---default
+---equipment
+---execute
+---exit
+---lineprobing
+---linerate
+---lines
+---meansq
+---targetmargin
+---wctargetmargin
```

"Management" configuration commands:

```
dsl-group
|
+---alarm
+---eoc
```

Additional configuration commands for inter-vendor compatibility:

```
dsl-group
|
+---4-wire-mode
+---efmaggregation
+---pmmsmode
+---tclayermode
+---vendorname
+---vendorspecoctets
```

2.3.3.5 DSL group configuration commands

Use the following command to specify the SHDSL standard that will be used.

```
CLI(dsl-group)> annex { A-F | B-G | auto } [tc-pam { 16 | 32 | auto }]
```

`annex` has the following possible values:

- `A-F`: the North-American SHDSL standard will be used.
- `B-G`: the European SHDSL standard will be used (default value).
- `auto`: this value is only possible for CPE devices; the annex is automatically set to the CO value.

`tc-pam` has the following possible values:

- `16`: with `tc-pam 16`, the line rate ranges from 192 kbps to 3840 kbps. This will automatically be selected when `caplist` is set to `oldstyle`; refer to the command below.
- `32`: with `tc-pam 32`, the line rate ranges from 768 kbps to 5696 kbps.
- `auto`: when using this value, the modulation will be determined automatically.

Note: For CPE devices, only `auto` is possible.

Use the following command to set which type of capability lists can be exchanged, default: `auto`.

During the SHDSL initial handshake phase, the two peers of the link exchange information on their capabilities. This includes whether or not they support G.SHDSL.bis. Some older DSLAM SHDSL cards refuse to proceed with the SHDSL handshake if they receive a capabilities list as defined in the newer G.SHDSL.bis standard. Therefore this command can be used to make the router interoperable with these older SHDSL implementations.

```
CLI(dsl-group)> caplist { oldstyle | newstyle | auto }
```

- `oldstyle`: the peer device only supports SHDSL (neither G.SHDSL.bis nor EFM).

- `newstyle`: the peer device supports SHDSL and G.SHDSL.bis ATM (not necessarily EFM).
- `auto`: (default) the OneOS-based router decides itself. This value is only valid for CPE devices. For CO devices, this value is internally translated to `newstyle`.

Use the following command to set the current condition startup margin, in function of which a line speed has to be selected during the ITU-T G.994.1 auto speed negotiation, default: `disabled`.

This command takes into account the current condition the device is operating under, during which the connection has to remain up. The higher the `cctargetmargin` will be, the lower the selected line speed will be, but the more stable the line will be.

If the `cctargetmargin` is disabled, the target margin is not considered during the ITU-T G.994.1 auto speed negotiation, i.e. all speeds are available (the target margin is the amount of received signal power in excess of that required to achieve the DSL target bit error rate of 10^{-7}).

```
CLI(dsl-group)> cctargetmargin { disabled | -10db..+10db }
```

For example, when setting `cctargetmargin` to 8 dB, there must be an 8 dB difference between signal and noise level under the current conditions, for the line to remain up.

If both `ccTargetMargin` and `wcTargetMargin` are disabled, no line probing will be done.

Use the following command to set the CRC error checking, default: 10 seconds

```
CLI(dsl-group)> crc { no | check [<crc-time-check>] }
```

`crc-time-check` is being any value in seconds between 1 and 30. It is expressed in *consecutive errored seconds*. This means, when set to 10, that after 10 consecutive errored seconds, the device will retrain.

In order to stop the CRC check, use the no form of the command:

```
CLI(dsl-group)> crc no check
```

Use the following command to set the equipment type, default: CPE.

```
CLI(dsl-group)> equipment { CO | CPE }
```

- `CO`: Use this value to set up the device as a Central Office type of equipment.
- `CPE`: Use this value to set up the device as a Customer Premises type of equipment.

Use the following command to set the line probing mechanism mode (applicable only in CO mode).

```
CLI(dsl-group)> lineprobing { adaptive | normal }
```

- `adaptive`: (default value) the adaptive line probing works on all lines together so, if one line goes down, it will retrain also all the other lines. This is done to guarantee that all lines will synchronize in a situation with a lot of inter line cross-talk. The adaptive line probing mechanism will lower the maximum speed and retrain all lines until a speed is found which is stable for all active lines.
- `normal`: use this mode to deactivate the adaptive line probing (making not all lines to retrain).

Use the following command to set the data transfer rate in fixed mode. The rate value is fixed between 192 and 5696 kbps in steps of 64 kbps (per line pair G.SHDSL.bis).

```
CLI(dsl-group)> linerate fixed <value>
```

Use the following command to set the range of bit rates that the G.SHDSL.bis chipset can use in adaptive mode. The device and DSLAM provide the highest line rate possible within the range defined here according to line features (length, attenuation) and tolerable Bit Error Ratio. Two parameters define the minimum rate and the maximum rate (in kbps).

```
CLI(dsl-group)> linerate adaptive <value1> <value2>
```

To use the default linerate setting (adaptive mode from 192 to 5696 kbps per line pair), proceed as follows:

```
CLI(dsl-group)> default linerate
```

Note: the line rate set with the above commands is the line rate for one pair. So, for instance, if 4 pairs are used, the total rate is multiplied by 4.

Use the following command to specify the lines that will be used. By default, the value `1-4` is taken, which means that line pair 1 till 4 are taken into account for the synchronization with the remote side.

```
CLI(dsl-group)> lines { 1 | 2 | 3 | 4 | 1-2 | 1-3 | 1-4 }
```

Note that for example: using 2 means that only line pair 2 is taken into account for synchronization with the remote side.

When using EFM, it is mandatory to configure line 1 as a member of the DSL-GROUP even if the line is not connected. If line 1 is not configured, the SHDSL status will show that the other lines are up, but no data traffic will go over the link.

Use the following command to set the minimum average noise margin threshold the line signal has to comply with. When not complying with these conditions, the device will retrain.

```
CLI(dsl-group)> meansq { no | check [<sq-time-check> [<sq-threshold>]] }
```

`sq-time-check` is used to set the signal quality time period (expressed in seconds), during which the noise margin is measured. Its range can be set between 1 and 30 seconds (default 10 seconds).

`sq-threshold` is used to set the minimum average noise margin threshold. Its range can be set between –2 dB and 10 dB (default +2 dB).

If the threshold value is exceeded during the check period, the device retrains.

Use the following command to specify how target margins are set (applicable only in CPE mode).

```
CLI(dsl-group)> targetmargin { normal | forced }
```

If set to `normal` (default value), the CPE will use the target margins (current condition and worst case) of the CO for line probing.

If set to `forced`, the CPE will ignore the CO configuration and use `cctargetmargin` and `wctargetmargin` for line probing.

Use the following command to set the worst condition startup margin, in function of which a line speed has to be selected during the ITU-T G.994.1 auto speed negotiation.

This command is similar to the `cctargetmargin` command (explained above), except that this command takes into account the worst possible condition that can occur on the line, during which the line still has to remain up.

```
CLI(dsl-group)> wctargetmargin { disabled | -10db..+10db }
```

For example, when setting `wctargetmargin` to 5 dB, there must be a 5 dB difference between signal and noise level for the line to remain up.

If both `cctargetMargin` and `wctargetMargin` are disabled, no line probing will be done.

Use the following command to make the OneOS-based router detecting the DSLAM configuration when setting up a connection; and then sets the appropriate parameters (unless some parameters are forced in the CLI configuration). This applies to the number of line pairs in use and possibly to some more parameters depending on the DSLAM chipset. This command is applicable only for ATM connections. It has no impact on EFM connections. It sets the connection mode used for the handshake. The handshake can be done in one or multi-pair mode. Then the CPE retrieves the connection parameters from the DSLAM and initializes the line pairs according to the retrieved parameters. The activate parameter determines the behavior of unused lines. By default only used lines will be activated, all other lines will remain silent. If set to all then also unused lines will be activated.

```
CLI(dsl-group)> autoconfig [multi-pair|single-pair] [activate {used|all}]
```

Note 1: The `autoconfig` mode will always try to synchronize on the first pair when configured in `single-pair`. If this line pair is not answering (e.g. it is not connected), the synchronization on the other line pairs will never be started (unless explicitly requested by `activate all` parameter).

Note 2: The `autoconfig` mode does not involve an automatic fallback mechanism to a smaller number of line pairs if one line pair disconnects for some reason. Unlike EFM or IMA it is inherent to the ATM encapsulation over SHDSL links that the complete link is broken if one line pair is no more operational (see `lineprobing` command).

To disable `autoconfig`:

```
CLI(dsl-group)> no autoconfig
```

2.3.3.6 DSL group monitoring configuration commands

Span monitoring is a feature of OneOS that allows the monitoring of G.SHDSL.bis repeaters and remote line termination. By default, the span monitoring is disabled. To enable span monitoring:

```
CLI(dsl-group)> eoc span-monitor enabled [delay <0-29>
                                         [interval { 30|60|90|120 }]]
```

The timing of the periodic status requests can be configured. The full quarter of an hour is used as a reference point. The requests are executed after an adjustable delay of (0-29, value in seconds). After the full quarter and periodically thereafter with the configured interval (30s, 60s, 90s or 120s), the status requests are sent. Default delay: 0 second. Default interval: 30 seconds.

To disable span monitoring:

```
CLI(dsl-group)> eoc span-monitor disabled
```

Some traps can be sent over SNMP to raise some alarms about service degradation (HDSL2-SHDSL-LINE-MIB). The traps are generated when certain thresholds are crossed. The first threshold defines the SNR (Signal-to-Noise Ratio) margin threshold in dB. When the current, estimated SNR is equal to or below the configured value, an alarm will be sent (hdsl2ShdslSNRMarginCrossing trap).

```
CLI(dsl-group)> alarm snrThreshold { disabled | enabled <-127..128> }
```

The next threshold configures the threshold for the number of Errored Seconds (ES) within any given 15-minute performance data collection interval. If the value of Errored Seconds in a particular 15-minute collection interval reaches/ exceeds this value, an hdsl2ShdslPerfESThresh trap is generated.

```
CLI(dsl-group)> alarm esThreshold { disabled | enabled <1..900> }
```

A Severely Errored Second (SES) is a one-second period that contains >30% errored blocks or at least one severely disturbed period. The next command is similar to **`esThreshold`**, but applies to SES:

```
CLI(dsl-group)> alarm sesThreshold { disabled | enabled <1..900> }
```

Then, the changes are validated via the following commands:

```
CLI(dsl-group)> execute
CLI(dsl-group)> exit
CLI(conf-ctrl)> exit
CLI(configure)>
```

To enable SNMP traps:

```
CLI(configure)> [no] snmp traps shdsl
```

2.3.3.7 Additional DSL group configuration commands

The following commands have been added for inter vendor compatibility.

The following parameter is only relevant for ATM operation in interoperability with Conexant (GlobeSpan) chipsets. Use the following command to set the operational mode of the lines, default: `auto`.

```
CLI(dsl-group)> 4-wire-mode { standard | enhanced | auto }
```

Note: when actually using less than 4 wires, this command can be ignored.

Use `standard` to have the handshake status exchanged in ATM 4-wire mode according to the SHDSL standard (exchange on the master wire pair only).

Use `enhanced` to have the handshake status exchanged in ATM 4-wire mode according to Conexant "enhanced" mode (exchange on all wire pairs). This was the only supported mode on Conexant chipsets with older software version (below 2.5 on Orion). Newer software versions support both standard and enhanced modes (configurable).

Use `auto` to have ATM 4-wire mode decided automatically based on the readout of the peer's chipset and software version. If the chipset is Conexant with an older software version, the enhanced mode is used. Otherwise the standard mode is used.

The following parameter is only applicable for EFM operation.

```
CLI(dsl-group)> efmaggregation { disabled | static | negotiated | auto }
```

Use `disabled` only for 1-pair operation; in this case the additional EFM aggregation layer is not present. This slightly optimizes the bandwidth use on a single pair.

Use `static` to have the number of lines used for EFM aggregation being exactly the number of line pairs configured. There is no negotiation with the peer SHDSL on which line pairs to bond. Use this option if the DSLAM does not support EFM negotiation of line pairs (e.g. Hatteras).

Use `negotiated` (default) to have, before the SHDSL handshake, the two SHDSL peers negotiate which line pairs will be used and bonded, based on configuration on both sides of the line pairs.

Use `auto` only when configured as central (CO) device. The central device decides which pairs to take into account for the bonding based on the peer's reaction. It can distinguish between different remote devices e.g. if a line pair would be wrongly connected.

The following command is important for compatibility between two devices of which one is equipped with a GlobeSpan chip set, more specifically with regards to the line probing process (PMMS stands for Power Measurement Modulation Session): when two such devices are interconnected, there is a compatibility issue with regard to the line probing timeout used by the chip sets.

Therefore, when the remote device is equipped with a GlobeSpan chip set, the `pmmsmode` must be set to GlobeSpan.

Note that this applies to GlobeSpan software versions 2.5.x and 3.0.x.

```
CLI(dsl-group)> pmmsmode { normal | globespan }
```

- `normal`: Use this value for normal operation.
- `globespan`: Use this value when the remote device is equipped with a GlobeSpan chip set.

The following parameter is relevant for EFM operation and indicates whether the DSLAM supports only G.SHDSL.bis capabilities or also EFM capabilities.

```
CLI(dsl-group)> tclayermode { single | all }
```

Use `single` (default) when the DSLAM supports the EFM capabilities.

Use `all` if the DSLAM does not support the EFM capabilities, but still supports EFM. This may occur with DSLAMs that use an older SHDSL chipset version that supports EFM but not yet the EFM capabilities (pre-standard) (e.g. Hatteras).

Note: This parameter is related to the `caplist` parameter, although it is different. `caplist oldstyle` means the DSLAM only supports the SHDSL standard. `caplist newstyle` means the DSLAM supports also the G.SHDSL.bis standard.

The following parameter allows setting which type of chipset the peer device is equipped with.

```
CLI(dsl-group)> vendorname { IFTN | GSPN }
```

Use `IFTN` if the peer device is equipped with an Infineon chipset.

Use `GSPN` if the peer device is equipped with a GlobeSpan octane chipset. Should be used together with:

```
pmmsmode gspn
vendorspecoctets 2535
caplist oldstyle
```

Note: When the `caplist` parameter is set to `auto` the OneOS-based router adapts `vendorname` automatically after detection of the peer device.

The following parameter is only relevant for ATM operation versus a DSLAM based on the Conexant Octane chipset with the software versions 5.4 or 5.6 (below version 5.7). When using the `autoconfig` command, the router reads the chipset and software version of the DSLAM and adapts itself the `vendorspecoctets`. The purpose of this parameter is only for unknown DSLAMs or chipsets/versions to make it interoperable without needing to change the firmware.

```
CLI(dsl-group)> vendorspecoctets <octet1> <octet2>
```

2.3.3.8 DSL group configuration default commands

Use the `default` command to restore the default settings of the following commands:

```
4-wire-mode
annex
caplist
cctargetmargin
crc
efmaggregation
equipment
lineprobing
linerate
lines
meansq
pmmsmode
targetmargin
tclayermode
vendorspecoctets
wctargetmargin
```

For example, the following command sets the CRC error checking back to the default value:

```
CLI(dsl-group)> default crc
CLI(dsl-group)>
```

2.3.3.9 Statistics and Traces

To activate traces, use the following command:

```
CLI> trace filter add sys hdsl all
```

Note: in the following statistics, some values are only relevant when the line is in DATA state. They may show non significant (rubbish) values when the line is not connected.

2.3.3.9.1 DSL group with ATM statistics

The DSL group statistics are viewed as follows:

```
CLI> show dsl-group 0

Current Status
 --------------
  Admin status             : UP
  Oper status              : UP
  Global Up/Down Counter   : 2
  Up time duration         : 0d 0h 1m 9s
  User Speed               : 22784kbps
  Used Lines               : 1,2,3,4

 User Configuration
 ------------------
  lines            : 1,2,3,4
  encapsulation    : atm
  equipment        : cpe (annex B-G TC-PAM auto)
  linerate         : adaptive / 192 kbps to 5696 kbps
  caplist          : auto
  ccTargetMargin   : 2dB
  wcTargetMargin   : disabled
  targetMargin     : normal
  4-wire-mode      : auto
  pmmsMode         : normal
  autoconfig       : enabled (multi-pair activate used)
  lineprobing      : adaptive
  tcLayerMode      : single
  vendorName       : IFTN
  vendorSpecOctets : 0000
  retrain criteria :
    crc check      : retrain on 3 consecutive errored sec
    meansq check   : retrain when noiseMargin lower than 2dB during 5 seconds

 Operational Mode
 ------------------

                            line 1         line 2         line 3         line 4
  Link status             : UP_DATA_MODE   UP_DATA_MODE   UP_DATA_MODE   UP_DATA_MODE
  Data rate               : 5696 kbps      5696 kbps      5696 kbps      5696 kbps
  Max Speed               : 5696 kbps      5696 kbps      5696 kbps      5696 kbps
  Negotiated Constellation: 32 TC-PAM      32 TC-PAM      32 TC-PAM      32 TC-PAM
  Used Capability List    : newstyle       newstyle       newstyle       newstyle
  Region Information      : annex B-G      annex B-G      annex B-G      annex B-G
  Line attenuation        :  0.0 dB         0.0 dB         0.0 dB         0.0 dB
  Noise margin            : 19.0 dB        19.0 dB        19.0 dB        19.0 dB
  Transmit power          :  8.5 dB         8.5 dB         8.5 dB         8.5 dB
  Power back-off          :  6.0 dB         6.0 dB         6.0 dB         6.0 dB
  Last Fail condition     : GHS_NO_COMMON_ NO_COND_       NO_COND_       NO_COND_
  Last Fail reason        : ERR_MS_NO_MODE_ ERR_UNUSED_    ERR_UNUSED_    ERR_UNUSED_
  Pmms Mode               : normal         normal         normal         normal
  Vendor Name             : IFTN           IFTN           IFTN           IFTN
  Vendor Spec Octets      : 0000           0000           0000           0000
  Pair Order              : master         slave3         slave1         slave2

 Remote info:
 ------------

  Vendor Name    : IFTN ( 49 46 54 4E)
  Vendor version : NA

 Retrain criteria:
 -----------------

        +-----------------------------------+-----------------------------+
        | SQ                                | CRC                         |
        +-----------------------------------+-----------------------------+
Pair 1 | average value in  5s is  18.98 dB |  0 consecutive errored sec  |
       +-----------------------------------+-----------------------------+
Pair 2 | average value in  5s is  18.98 dB |  0 consecutive errored sec  |
       +-----------------------------------+-----------------------------+
Pair 3 | average value in  5s is  18.98 dB |  0 consecutive errored sec  |
       +-----------------------------------+-----------------------------+
Pair 4 | average value in  5s is  18.98 dB |  0 consecutive errored sec  |
       +-----------------------------------+-----------------------------+

Retrain counters:
-----------------
```

```
       +-----------------------------------------------------------------------------------------------------------+
       | Nb retrains since power-on                          | Nb retrains since last read                         |
       +-----------------------------------------------------------------------------------------------------------+
       | SQ     | CRC    | HEC    | Cell-D | Spont  | User   | SQ     | CRC    | HEC    | Cell-D | Spont  | User   |
       +-----------------------------------------------------------------------------------------------------------+
Total  |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |
       +-----------------------------------------------------------------------------------------------------------+
Pair 1 |       0 |       0 |     n/a |     n/a |     n/a |     n/a |       0 |       0 |     n/a |     n/a |     n/a |     n/a |
       +-----------------------------------------------------------------------------------------------------------+
Pair 2 |       0 |       0 |     n/a |     n/a |     n/a |     n/a |       0 |       0 |     n/a |     n/a |     n/a |     n/a |
       +-----------------------------------------------------------------------------------------------------------+
Pair 3 |       0 |       0 |     n/a |     n/a |     n/a |     n/a |       0 |       0 |     n/a |     n/a |     n/a |     n/a |
       +-----------------------------------------------------------------------------------------------------------+
Pair 4 |       0 |       0 |     n/a |     n/a |     n/a |     n/a |       0 |       0 |     n/a |     n/a |     n/a |     n/a |
       +-----------------------------------------------------------------------------------------------------------+
```

```
Line Statistics:
----------------
```

```
       +-----------------------------------------------------------------------------------------------------------+
       | Nb since power-on                                   | Nb since last read                                  |
       +-----------------------------------------------------------------------------------------------------------+
       | LL     | CV     | ES     | SES    | LOSWS  | UAS    | LL     | CV     | ES     | SES    | LOSWS  | UAS    |
       +-----------------------------------------------------------------------------------------------------------+
Pair 1 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |
       +-----------------------------------------------------------------------------------------------------------+
Pair 2 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |
       +-----------------------------------------------------------------------------------------------------------+
Pair 3 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |
       +-----------------------------------------------------------------------------------------------------------+
Pair 4 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |
       +-----------------------------------------------------------------------------------------------------------+
```

```
Utopia Statistics:
------------------

Cell delineation: sync

        (power-on  lastread)
 CRC           0         0
 RxCells       0         0
 RxDrops       0         0
 RxHec         0         0
 TxCells       0         0
```

2.3.3.9.2 DSL group with EFM statistics

The DSL group statistics are viewed as follows:

```
CLI> show dsl-group 0

Current Status
 --------------
  Admin status             : UP
  Oper status              : UP
  Global Up/Down Counter   : 1
  Up time duration         : 0d 0h 8m 43s
  User Speed               : 17088kbps
  Used Lines               : 1,2,3,4

 User Configuration
 ------------------
  lines            : 1,2,3,4
  encapsulation    : efm
  equipment        : cpe (annex B-G TC-PAM auto)
  linerate         : adaptive / 192 kbps to 5696 kbps
  caplist          : auto
  ccTargetMargin   : 2dB
  wcTargetMargin   : disabled
  targetMargin     : normal
  4-wire-mode      : auto
  pmmsMode         : normal
  autoconfig       : enabled (multi-pair activate used)
  lineprobing      : adaptive
  active link nr   : 1
  active user speed: 0
 efmAggregation   : negotiated
 tcLayerMode      : single
 vendorName       : IFTN
 vendorSpecOctets : 0000
 retrain criteria :
   crc check      : retrain on 3 consecutive errored sec
   meansq check   : retrain when noiseMargin lower than 2dB during 5 seconds

Operational Mode
------------------

                           line 1         line 2         line 3         line 4
 Link status             : UP_DATA_MODE   DOWN_READY     UP_DATA_MODE   UP_DATA_MODE
 Data rate               : 5696 kbps      0    kbps      5696 kbps      5696 kbps
 Max Speed               : 5696 kbps      5696 kbps      5696 kbps      5696 kbps
 Negotiated Constellation: 32 TC-PAM      unknown        32 TC-PAM      32 TC-PAM
 Used Capability List    : newstyle       unknown        newstyle       newstyle
 Region Information      : annex B-G      annex unknown  annex B-G      annex B-G
 Line attenuation        :  0.0 dB         0.0 dB         0.0 dB         0.0 dB
 Noise margin            : 19.0 dB        -7.0 dB        19.0 dB        19.0 dB
 Transmit power          :  8.5 dB         0.0 dB         8.5 dB         8.5 dB
 Power back-off          :  6.0 dB         0.0 dB         6.0 dB         6.0 dB
 Last Fail condition     : NO_COND_       NO_COND_       NO_COND_       NO_COND_
 Last Fail reason        : ERR_UNUSED_    ERR_UNUSED_    ERR_UNUSED_    ERR_UNUSED_
 Pmms Mode               : normal         normal         normal         normal
 Vendor Name             : IFTN           IFTN           IFTN           IFTN
 Vendor Spec Octets      : 0000           0000           0000           0000
 Pair Order              : master         unknown        master         master

Remote info:
------------

 Vendor Name    : IFTN ( 49 46 54 4E)
 Vendor version : NA

Retrain criteria:
-----------------

       +-----------------------------------+-----------------------------+
       | SQ                                | CRC                         |
       +-----------------------------------+-----------------------------+
Pair 1 | average value in  5s is  18.98 dB |  0 consecutive errored sec  |
       +-----------------------------------+-----------------------------+
Pair 2 | average value in  5s is   1.16 dB |  0 consecutive errored sec  |
       +-----------------------------------+-----------------------------+
Pair 3 | average value in  5s is  18.98 dB |  0 consecutive errored sec  |
       +-----------------------------------+-----------------------------+
Pair 4 | average value in  5s is  18.98 dB |  0 consecutive errored sec  |
       +-----------------------------------+-----------------------------+

Retrain counters:
-----------------

       +-----------------------------------------------------------------------------------------------------------+
       | Nb retrains since power-on                          | Nb retrains since last read                         |
       +-----------------------------------------------------------------------------------------------------------+
       | SQ     | CRC    | HEC    | Cell-D | Spont  | User   | SQ     | CRC    | HEC    | Cell-D | Spont  | User   |
       +-----------------------------------------------------------------------------------------------------------+
Total  |      5 |      0 |      0 |      0 |      0 |      0 |      1 |      0 |      0 |      0 |      0 |      0 |
       +-----------------------------------------------------------------------------------------------------------+
Pair 1 |      1 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |
       +-----------------------------------------------------------------------------------------------------------+
Pair 2 |      2 |      0 |      0 |      0 |      0 |      0 |      1 |      0 |      0 |      0 |      0 |      0 |
       +-----------------------------------------------------------------------------------------------------------+
Pair 3 |      1 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |
       +-----------------------------------------------------------------------------------------------------------+
Pair 4 |      1 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |
       +-----------------------------------------------------------------------------------------------------------+

Line Statistics:
----------------

       +-----------------------------------------------------------------------------------------------------------+
       | Nb since power-on                                   | Nb since last read                                  |
       +-----------------------------------------------------------------------------------------------------------+
       | LL     | CV     | ES     | SES    | LOSWS  | UAS    | LL     | CV     | ES     | SES    | LOSWS  | UAS    |
       +-----------------------------------------------------------------------------------------------------------+
Pair 1 |      1 |      0 |      1 |      1 |      5 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |
       +-----------------------------------------------------------------------------------------------------------+
Pair 2 |      1 |      0 |      1 |      1 |     10 |      0 |      0 |      0 |      0 |      0 |      5 |      0 |
       +-----------------------------------------------------------------------------------------------------------+
Pair 3 |      1 |      0 |      1 |      1 |      5 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |
       +-----------------------------------------------------------------------------------------------------------+
Pair 4 |      1 |      0 |      1 |      1 |      5 |      0 |      0 |      0 |      0 |      0 |      0 |      0 |
       +-----------------------------------------------------------------------------------------------------------+
```

2.3.3.9.3 DSL group monitoring statistics

The EOC Span monitoring statistics are viewed as follows.

The following command shows inventory details about the whole span (CO, CPE & repeaters). This information is also present in the hdsl2ShdslInventoryTable in HDSL2-SHDSL-LINE-MIB DEFINITIONS.

```
CLI> show dsl-group 0 inventory
```

The following command shows for all devices in the span the current status info.

```
CLI> show dsl-group 0 span-status
```

These pieces of information correspond to the following MIB tables:

- hdsl2ShdslEndpointCurrTable (partially, no counters for ES, SES, CRC ...)
- hdsl2ShdslEndpointMaintTable
- hdsl2ShdslUnitMaintTable

The following information is present:

- line attenuation
- SNR margin
- Power Back Off (PBO)
- tip-ring reversal
- activation state
- alarm status (SNR alarm, ES alarm, SES alarm)
- power source (local, span)

The following command shows for all devices in the span the global counters since the last restart and for the current 15 min and 1-day interval. The information is similar to the following MIB table: hdsl2ShdslEndpointCurrTable (only the counters; ES, SES, CRC ...).

```
CLI> show dsl-group 0 span-counters
```

The following command shows for all devices in the span the counters for the 16 last 15-min intervals (ES, SES, CRC, LOSW and UAS). This corresponds to MIB table hdsl2Shdsl15MinIntervalTable.

```
CLI> show dsl-group 0 span-counters 15min
```

The following command shows for all devices in the span the counters for the 3 last 1-day intervals (ES, SES, CRC, LOSW and UAS). This corresponds to MIB table hdsl2Shdsl1DayIntervalTable.

```
CLI> show dsl-group 0 span-counters 1day
```

2.3.3.10 G.SHDSL.bis configuration example

The following listing shows a G.SHDSL.bis configuration example of an One100D.

It shows the use of the `controller shdsl 0` and `dsl-group` commands, and the use of the DSL group in an ATM interface:

```
configure terminal
no reboot recovery-on-error
logging buffered size 16364
controller shdsl 0
  dsl-group 0
    caplist oldstyle
    4-wire-mode standard
    pmmsmode globespan
    execute
  exit
exit
set multiple-conf-sessions enable
hostname ONE100D
interface FastEthernet 1/0
 ip address 1.1.1.2 255.255.255.252
exit
interface atm 0
 driver ident 0
  execute
 exit
 dsl-group 0
exit
interface atm 0.1
 pvc ipoa vpi 0 vci 32
  ip address 3.3.3.1 255.255.255.252
  inarp client
  qos vbr pcr 4608000 scr 64000 mbs 65535
  execute
 exit
exit
ip route 2.2.2.0 255.255.255.252 3.3.3.2
no snmp set-write-community private
no snmp set-read-community public
voice-default
```

2.3.4 Inverse Multiplexing for ATM (IMA) over G.SHDSL.bis

2.3.4.1 Principles of IMA

They are described in the G.SHDSL section. Please refer to 2.3.2.1.

2.3.4.2 Configuration

First, set the common physical layer parameters for G.SHDSL.bis links. This is achieved by configuring the G.SHDSL.bis DSL group as described in 2.3.3.

Then enter in IMA configuration mode under ATM interface configuration mode:

```
CLI(configure)> interface atm <interface-number>
CLI(config-if)> [no] ima
CLI(ima)>
```

`interface-number` is always 0.

The CLI has entered into the IMA configuration mode. The parameters related to IMA itself are the same than for IMA over G.SHDSL. Please refer to 2.3.2.2 for more information.

Then, the DSL group is applied to the IMA and the IMA configuration is terminated.

```
CLI(ima)> dsl-group 0
CLI(ima)> execute
CLI(ima)> exit
CLI(config-if)>
```

Last, the ATM interface configuration is terminated.

```
CLI(config-if)> exit
CLI(configure)>
```

2.3.4.3 Statistics

Please refer to 2.3.2.3.

2.3.4.4 Example

The following example is an IMA interface with G.SHDSL.bis links:

```
controller shdsl 0
 dsl-group 0
  autoconfig
  execute
  exit
 exit
interface atm 0
 ima
  dsl-group 0
  execute
  exit
 exit
```

2.3.5 E1/T1 ATM Configuration

The E1 uplink interface of the ONE60/200 routers supports ATM cell transport. The implementation is fully compatible with the ATM forum specifications, where the E1 can be fully used at 1920 kbps (30 timeslots) or partially. 'Partial E1' is referred to as Nx64 mode and means that a number of time slots can form a time slot bundle, which can be used for transport of ATM cells. However, the partial E1 does not support several time slot bundles (NxPx64). For details about the specifications, the user should refer to the ATM Forum specifications:

- Full E1: AF-PHY-0064, available under:
  http://broadband-forum.org/ftp/pub/approved-specs/af-phy-0064.000.pdf
- Partial E1: AF-PHY-0130, available under:
  http://broadband-forum.org/ftp/pub/approved-specs/af-phy-0130.000.pdf
- T1: AF-PHY-0016, available under:
  http://broadband-forum.org/ftp/pub/approved-specs/af-phy-0016.000.pdf

The configuration commands are the following to create the E1/T1 ATM interface:

```
CLI> configure terminal
CLI(configure)> interface atm <atm-if>
CLI(config-if)> e1
CLI(config-if)> t1
```

`atm-if` is an arbitrary identifier (starting from 0) that identify a physical interface where ATM cells are supported. The E1 interface can be master (the device provides its internal clock) or slave (clock received).

```
CLI(config-if-e1)> equipment { master | slave }
```

Then the time slots of the bundle must be entered. Note that the time slots do not need to be contiguous. The configuration commands are:

```
CLI(config-if-e1)> ts <first-ts-1>[-<last-ts-1>]
CLI(config-if-e1)> ts <first-ts-2>[-<last-ts-2>]
```

Note that the use of time slot #0 and #16 are forbidden by the ATM specification for E1. For T1, useable time slots range from 1 to 24. Example for Time Slots:

```
CLI(config-if-e1)> ts 1-15
CLI(config-if-e1)> ts 17
CLI(config-if-e1)> ts 19-23
```

To enable the scrambling on the E1 interface, use the following command:

```
CLI(config-if-e1)> serial-scramble
```

The default value is "`no serial-scramble`"; "`serial-scramble`" is the recommended value.

If you wish to activate the "coset" Header Error Correction, use the following command:

```
CLI(config-if-e1)> serial-heccoset
```

The default value is "`no serial-heccoset`".

To conclude the E1 ATM interface configuration, use the following commands:

```
CLI(config-if-e1)> execute
CLI(config-if-e1)> exit
CLI(config-if)> exit
CLI(configure)>
```

The created ATM interface can be used with conventional ATM commands, which are common to other ATM-based interfaces such as G.SHDSL and ADSL.

2.3.5.1 Configuration Example

This example creates an E1 ATM interface with full line speed. An ATM PVC is created at the maximum line rate and supports a simple PPPoA connection:

```
configure terminal
 interface atm 0
  e1
   ts 1-15
   ts 17-31
   equipment slave
   execute
  exit
 exit
 interface atm 0.1
  pvc pppoa vpi 0 vci 32
   ip address 10.10.10.1
   ipcp static
   authentication no
   qos cbr pcr 1920000
   execute
  exit
 exit
exit
```

2.3.6 Inverse Multiplexing for ATM (IMA) over E1/T1

2.3.6.1 Principles of IMA

IMA is a technique enabling to split and reassemble an ATM cell stream over multiple physical links. It was defined by the ATM Forum recommendation AF-PHY-0086.0001. For more details, please refer to:

`http://broadband-forum.org/ftp/pub/approved-specs/af-phy-0086.001.pdf`

This technique is highly efficient to increase incrementally the capacity of transmission links. The ONE400 can be equipped with an IMA module supporting 4 or 8 E1/T1. Such module fills the bandwidth gap between a single, 2 Mbps E1 link and the E3 (34 Mbps) for a fraction of the E3 bandwidth cost.

An IMA interface forms an IMA group. An IMA group is actually made up of several physical links. The role of IMA is to split the incoming cell traffic over the different physical interfaces. The IMA group must respectively reassemble the cell stream at the remote end. The IMA algorithm ensures the cell stream is reassembled in the proper order and compensates for possible inter-link delays. An embedded OAM ensures links can be dynamically added / removed in the group.

The data transiting over the E1/T1 links are made up of:

- ATM cells (the cells are sent over each link on a cell-by-cell basis)
- ICP (IMA Control Protocol) cells
- Filler cells (when no ATM cell have to be sent)

The IMA group manages the cell stream over the N links. Over each link, the IMA group controls the cell stream by using an IMA frame of length M. An IMA frame contains one ICP cell, some ATM cells and/or filler cells. The IMA frame is the unit of control of IMA.

The ATM, filler and ICP cells are transported over E1/T1 in a similar manner as ATM over E1/T1. This means:

- For E1: data are transmitted over 30 time slots (64 kbps, time slot 0 and 16 excluded). An ATM actual bit rate of 1920*N*(M-1)/M kbps is offered (we must remove the ICP cell).
- For T1: data are transmitted over 24 time slots (64 kbps, time slot 0 excluded). An ATM actual bit rate of 1536*N*(M-1)/M kbps is offered (we must remove the ICP cell).

2.3.6.2 Configuration

To provision the IMA interface, first enter in ATM interface configuration mode:

```
CLI(configure)> interface atm <interface-number>
CLI(config-if)> [no] ima
CLI(config-ima)>
```

`interface-number` is usually 0.

The CLI has entered into the IMA configuration mode. You can set the IMA frame length M, when typing the following command:

```
CLI(config-ima)> frame-length { 32 | 64 | 128 | 256 }
```

The frame length must be the same at each end of the IMA link (default: 32 cells per frame). The following command sets the number of active links required to consider the interface as up and running (default: 1 link):

```
CLI(config-ima)> active-link-number <1..7>
```

Inter-link differential delays are the difference between transit delays of each link. The IMA group must compensate such differential delays till a maximum configured as follows:

```
CLI(config-ima)> max-delay <milliseconds>
```

The default value is 20 ms and the range of allowed delays is:

```
- <0..220> for E1
- <0..275> for T1
```

The following command sets the clock property for links of the IMA group:

```
CLI(config-ima)> clock-mode { dependent | independent }
```

If `dependent` is selected, it means that all links are synchronized on the same clock source; `independent` indicates that links may have different sources. Usually, all links are always synchronized on the same source; the default value is then `dependent`.

Then you must configure each E1/T1 link. Note that E1 and T1 cannot be simultaneously supported. The following command makes the CLI enter in PRI interface configuration mode:

```
CLI(config-ima)> [no] pri-link 2/<0..7>
CLI(config-pri)>
```

The PRI port number actually ranges from 0 to 3 or 0 to 7 depending on the number of links physically available on the IMA card you are using.

The following command configures the AIS mode (ITU-T G.775 or ETSI 300 233):

```
CLI(config-pri)> ais-mode { itu | etsi }
```

Default: `itu`.

The following command sets the clock mode (synchronized on the network = slave):

```
CLI(config-pri)> equipment { master | slave }
```

Default: `master`. When two devices function back-to-back, one must be master, the other is slave. The following command sets the E1/T1 interface in short or long haul (enabling higher reach):

```
CLI(config-pri)> haul { short | long }
```

Default: `short`. Then, configure the interface in E1 or T1 mode:

```
CLI(config-pri)> port-type { e1 | t1 }
```

For E1, you can set the following interface parameters:

```
CLI(config-e1)> framing { nil | ddf | mff-crc4 | mff-crc4-ext }
CLI(config-e1)> line-code { ami | hdb3 }
```

The framing attributes are:

- `nil` - Transparent Mode
- `ddf` - Double Frame Format
- `mff-crc4` - CRC4 Multi Frame Format
- `mff-crc4-ext` - CRC4 Multi Frame Format Extended G.706B

Default: `ddf` and `hdb3`.

For T1, you can set the following interface parameters:

```
CLI(config-t1)> framing { nil| f4| f12| esf| esf-crc6| esf-crc6-jt| f72 }
CLI(config-t1)> line-code { ami | b8zs }
```

The framing attributes are:

- `nil` - Transparent Mode (FALC acts as LIU only)
- `f4` - 4-Frame multiframe
- `f12` - 12-Frame multiframe (D4)
- `esf` - Extended Super Frame without CRC6
- `esf-crc6` - Extender Super Frame with CRC6
- `esf-crc6-jt` - Extended Super Frame with CRC6 JT G.706 for Japan
- `f72` - 72-Frame multiframe (SLC96)

Default: `nil` and `ami`.

Then enter `exit` to finish the link configuration and configure the following links. Then, enter `execute`.

2.3.6.3 Statistics

IMA statistics are obtained using the following command:

```
CLI> show ima
```

2.3.6.4 Example

The following example is an IMA interface with a single E1 link:

```
interface atm 0
ima
pri-link 2/0
port-type e1
exit
execute
exit
exit
```

2.3.7 ADSL

2.3.7.1 ADSL Configuration

To start the ADSL configuration, the user must enter into the configuration mode using the command:

```
CLI> configure terminal
```

First of all, the ADSL physical interface (the modem) must be configured.

The following command enters into the physical ADSL configuration mode (`if-number` must be set to 0):

```
CLI(configure)> interface adsl <if-number>
CLI(config-if)>
```

Then, specify the modem mode and country (optional).

The `AUTO` value is used to try to connect first using `GDMT` mode then using `ANSI_T1413 (POTS)` or `ETSI_DTS/TM_06006` (ADSL over ISDN).

```
CLI(config-if)> modem mode { AUTO | GDMT | GDMT_B | GDMT_B_DT
                           | ANSI_T1413 | ETSI_DTS/TM_06006 | GLITE
                           | G992_3A | G992_3B | G992_3J | G992_3M
                           | G992_5A | G992_5B | G992_5J | G992_5M }
```

The value for country is an integer value described in ITU-T T.35 recommendation (procedure for allocation of ITU-defined codes for non-standard facilities). For instance: France is 61, Germany is 66, UK is 180, USA is 181, and Belgium is 15.

```
CLI(config-if)> country <country-id>
```

In same cases a different ADSL firmware has to be used. Refer to the Software Release Note to know which ADSL firmware releases are available.

```
CLI(config-if)> firmware-alternate <0-1>
```

The following command is specific to ONE20/100 routers. **It is only for debug purpose**; it should not be used in products installed in live networks. `enabled` forces the modem to stay in level-3 power mode (the power-save mode will not therefore be used). The default is `disabled` (which means the modem can go to power mode level 0).

```
CLI(config-if)> l3-power-mode { disabled | enabled }
```

The following command must be used only when interoperability problem occurs. For OneOS-based routers fitted with Amazon chipset, it is now possible to set and clear the Lantiq interoperability bits of the Amazon bit pump (this command can also for be used for debug purpose).

```
CLI(config-if)> [no] interop { info_103_0 | info_103_1 | info_103_20
| info_103_23 | info_108_0 | optn_25_0 } set <bitmask> clear <bitmask>
```

`bitmask` is a 16-bit value which can be entered either in decimal or hexadecimal (e.g. 20 or 0x14).

To remove the interoperability command, use the no form of the above-referenced command.

Example for interoperability with an UTStarcom DSLAM in annex B mode:

```
interop info_103_0 set 0x0100 clear 0x0000
interop info_103_1 set 0x0100 clear 0x0000
```

The `execute` command closes the physical interface configuration putting into place the parameters that have been set.

```
CLI(config-if)> execute
```

Example:

```
interface adsl 0
```

```
modem mode AUTO
country 61
execute
exit
```

The next command removes the ADSL physical interface:

```
CLI(configure)> no interface adsl 0
```

The second step is to configure the ATM physical interface to use an ADSL logical channel.

The `adsl` command configures the physical interface for the ATM interface in ADSL mode; first, enter the ATM interface configuration mode:

```
CLI(configure)> interface atm <interface-number>
```

`interface-number` must be set to 0 for ONE60/200.

The CLI enters the ADSL configuration mode, when typing the next command:

```
CLI(config-if)> [no] adsl
CLI(adsl)>
```

Use the no form of the command to remove the ADSL interface:

Some parameters must be chosen before starting the ADSL interface. The related commands and parameters are:

```
CLI(adsl)> channel type <type>
```

This command specifies kind of ADSL channel we want to use with that ATM interface. The values could be `default`, `fast` or `interleaved`. However, the only available value today is `default` (`fast` or `interleaved` value are for future use).

```
CLI(adsl)> execute
```

The command closes the physical interface configuration putting into place the parameters that have been set.

Example:

```
interface atm 0
 adsl
  channel type default
  execute
 exit
exit
```

2.3.7.2 ADSL Statistics

```
CLI> show ADSL
Modem software version (0x3916): 3.9.22
(firmware version)
Modem in Showtime
(modem state)
Upstream Bitrate Fast: 0 kbps
(far-end fast channel bit rate)
Downstream Bitrate Fast: 0 kbps
(near-end fast channel bit rate)
Upstream Bitrate Interleaved: 160 kbps
(far-end interleaved channel bit rate)
Downstream Bitrate Interleaved: 608 kbps
(near-end interleaved channel bit rate)
Upstream bit rate occupation: 23
(% of the maximum attainable (Bmax))
Downstream bit rate occupation: 12
(% of the maximum attainable (Bmax))
Downstream Noise Margin: (67) 33.5 dB
(amount of increased noise that can be tolerate while maintaining a downstream BER of 10-7)
(<read value>) <decoded value>  dB
Upstream Noise Margin: (62) 31 dB
(amount of increased noise that can be tolerate while maintaining a upstream BER of 10-7)
(<read value>) <decoded value>  dB
Downstream Transmit Power: (40) 20 dBm
```

```
(ATU-C transmit power level)
(<read value>) <decoded value>  dBm
Upstream Transmit Power: (24) 12 dBm
(ATU-R transmit power level)
(<read value>) <decoded value>  dBm
Downstream Attenuation: (82) 41 dB
(difference between the power received by the ATU-R and a reference ATU-C transmit power
level)
(<read value>) <decoded value> dB
Upstream Attenuation: (58) 29 dB
(difference between the power received by the ATU-C and a reference ATU-R transmit power
level)
(<read value>) <decoded value>  dB
Near-end frame received with FEC (non interleaved): 0
(Number of near-end received superframes with one or more Reed-Solomon corrections in a data
frame)
Far-end frame received with FEC (non interleaved): 0
(Number of far-end received superframes with one or more Reed-Solomon corrections in a data
frame)
Near-end frame received with FEC (interleaved): 1
(Number of near-end received superframes with one or more Reed-Solomon corrections in a data
frame)
Far-end frame received with FEC (interleaved): 0
(Number of far-end received superframes with one or more Reed-Solomon corrections in a data
frame)
Near-end frame received with CRC error (non interleaved): 0
(Number of near-end received superframes with an incorrect CRC)
Far-end frame received with CRC error (non interleaved): 0
(Number of far-end received superframes with an incorrect CRC)
Near-end frame received with CRC error (interleaved): 0
(Number of near-end received superframes with an incorrect CRC)
Far-end frame received with CRC error (interleaved): 2
(Number of far-end received superframes with an incorrect CRC)
Near-end frame received with HEC error (non interleaved): 0
(Number of near-end received superframes with an HEC error)
Far-end frame received with HEC error (non interleaved): 0
(Number of far-end received superframes with an HEC error)
Near-end frame received with HEC error (interleaved): 0
(Number of near-end received superframes with an HEC error)
Far-end frame received with HEC error (interleaved): 0
(Number of far-end received superframes with an HEC error)
Near-end bitmap status: 00
(Current defects bitmap)
<value> (<value decoded in non-zeroe>)
Far-end bitmap status: 00
(Current defects bitmap)
<value> (<value decoded in non-zero >)
Near-end bitmap change: 00
(Defects bitmap, indicating changes)
<value> (<value decoded in non-zero >)
Far-end bitmap change: 00
(Defects bitmap, indicating changes)
<value> (<value decoded in non-zero >)
Near-end bitmap 2.5 s: 00
(Defects bitmap during 2.5S)
<value> (<value decoded in non-zero >)
Far-end bitmap 2.5 s: 00
(Defects bitmap during 2.5S)
<value> (<value decoded in non-zero >)
Carrier Load:
(Number of bits per symbol each tone carries (two tones per byte))
000 - 00 00 00 00 22 23 33 44 44 44 44 44 43 32 22 00 00 00 00 00
020 - 00 00 22 22 33 44 44 44 44 44 44 24 44 44 43 43 44 44 44 44
040 - 44 44 44 44 22 22 23 33 33 33 33 23 22 22 00 00 00 00 00 00
060 - 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
080 - 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
100 - 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
120 - 00 00 00 00 00 00 00 00
Rx Counter Fast Channel: 0
Rx Counter Interleaved Channel: 1069711
Tx Counter Fast Channel: 0
Tx Counter Interleaved Channel: 48079
Operational Mode (INFO_0): 2 (ITU G.992.1-ANNEX-A)
<value> (<decoded value>)
Operational Mode (INFO_1): 0 (NA)
<value> (<decoded value >)
Near-end Version: 0x21 AVN 1 IVN 0x01
<value> (<decoded value >
Far-end Version: 0xff (NA)
<value> (<NA meaning not applicable if 0xFF>)
Near-end Vendor: 0x0022
<value> (<NA meaning not applicable if 0xFF>)
Far-end Vendor: 0xffff (NA)
<value> (<NA meaning not applicable if 0xFFFF>)
UP-DOWN Counter: 0
ShowTime duration: 0d 0h 24m 27s
```

Example result for modem not connected:

```
Modem software version (0x3916): 3.9.22
Modem trying to connect
UP-DOWN Counter: 1
```

Values for modem state:

`Modem is IDLE`: waiting for an open command.

`Modem is LISTENING`: in listening state (only for Listen Round Robin (ATU-C) - not used).

`Modem trying to connect`: open command received and waiting for response.

`Modem is HANDSHAKE`: G.hs of ITU-T ACTIVATING phase.

`Modem connected`: ACT/ACK phase passed (ANSI) or handshaking phase terminated (ITU).

`Modem in Showtime`: ACTIVATING phase passed.

`Modem in unknown state`

Values for Defects bitmap `<decoded value>`:

```
Displayed using (<decoded value 1> | ... | <decoded value n>)
LOS: Loss of signal
LOF: Loss of Frame
LOP: Loss of power
LCD_NI: Loss of cell delineation (non interleaved)
LCD_I: Loss of cell delineation (interleaved)
LOM: Loss of margin
```

Values for Operational Mode (INFO_0) `<decoded value>`:

```
ANSI T1.413
ITU G.992.1-ANNEX-A
ITU G.992.2-ANNEX-A
```

Values for Operational Mode (INFO_1) `<decoded value>`:

```
ANSI ETSI DTS/TM-06006
ITU G.992.1-ANNEX-B
NA
```

Values for Near | Far-end Version `<decoded value>`:

```
NA
AVN: ANSI version Number
IVN: ACTIVATING Version Number
```

2.3.7.3 ADSL/VDSL daughterboard management

On selected OneOS-based routers the same RJ11 connector can be used for either ADSL or VDSL.

Please refer to chapter 3.3.4.4 for more information.

2.3.8 SDSL 2B1Q Configuration

SDSL 2B1Q should not be mixed up with the SHDSL standard (G.991.2).

To start the SDSL interface configuration, enter the following command from the global configuration terminal:

```
CLI(configure)> interface atm 0
CLI(config-if)> sdsl
CLI(sdsl)>
```

You can configure several parameters:

To set the connection mode, use the following command. `cpe` is the default value. When using `co`, the interface is configured so that it emulates the DSLAM interface. It can be useful for point-to-point connections.

```
CLI(sdsl)> equipment { cpe | co }
```

To set the framing type, use the following command:

```
CLI(sdsl)> framing { CLRCH | G991 | DLCC | GENERIC }
```

- `CLRCH`: ATM cells and no framing
- `G991`: G991.1 framing specification
- `DLCC`: DLCC framing specification
- `GENERIC`: no framing

To set the connection mode, use the following command:

```
CLI(sdsl)> connect-mode { BASIC | FMODE | COPPER | FAST }
```

- `BASIC`: basic operating mode of the chipset
- `FMODE`: mode compatible with Flow Point Corporation
- `COPPER`: mode compatible with Copper Mountain Networks Inc.
- `FAST`: GlobeSpan-specific: each side can select different data rates and a successful startup will occur at the lower of the two rates.

Then, you can select the line rate, where the interface speed can be negotiated between a minimum and maximum value or is fixed. The configuration syntax is:

```
CLI(sdsl)> linerate { fixed <speed>
                    | adaptive <min-speed> <max-speed>
                    | single <min-speed> <max-speed>
                    | fast <min-speed> <max-speed> }
```

The allowed speed values are provided in kilobit per second and must be provided as: `144+N*64 kbps` with N ranging from 0 to 34 (minimum: 144 kbps, maximum: 2320 kbps).

Details about non-fixed modes:

- `linerate adaptive`: (auto baud) negotiation between min rate and max rate
- `linerate single`: (single baud) startup is made at the lowest rate and using EOC, both sides negotiate a new rate
- `linerate fast`: same as adaptive, except that fewer startups are used on activation

Then, enter `execute` to validate the parameters and then `exit` two times.

To activate traces, use the following command:

```
CLI> trace filter add sys sdsl all
```

To show the SDSL interface status, use the following command:

```
CLI> show sdsl
```

2.3.8.1 Recommended Values Example for the Lucent Stinger DSLAM

```
interface atm 0
sdsl
equipment cpe
framing CLRCH
connect-mode FMODE
linerate fixed 2320
execute
exit
exit
```

2.3.9 STM-1

The ONE400 can be equipped with an STM-1 interface with optical fiber interface, supporting only ATM.

2.3.9.1 Configuration

To provision the STM-1 interface, first enter in ATM interface configuration mode:

```
CLI(configure)> interface atm <interface-number>
CLI(config-if)>
```

Note: `interface-number` is usually 0.

The CLI enters into the STM-1 configuration mode (`config-if`), when typing the following command:

```
CLI(config-if)> stm1
CLI(stm1)>
```

You can define the OAM behavior of the STM-1. You can enable or disable the emission of AIS (F3 OAM, not ATM), when the STM-1 interface is considered as down:

```
CLI(stm1)> [no] ais-shut
```

In case of poor transmission quality, the Bit Error Ratio (BER) is excessive. Two thresholds are defined so that the OAM considers the interface as having a degraded signal or failing signal. For the threshold configuration measuring degraded signal, the command is the following:

```
CLI(stm1)> sd-ber <3..9>
```

The parameter is the 10-exponent of the BER. For example, when the value is 8, it means the signal is considered as degraded when the BER has exceeded 10-8 (default value: 6)

For failure threshold, the command is the following:

```
CLI(stm1)> sf-ber <3..9>
```

`sf-ber` has the same type of value as `sd-ber.` Default value: 3.

If you wish to log alarms, use the report command as follows:

```
CLI(stm1)> [no] report { all
                       | rsop-oof
                       | rsop-lof
                       | rsop-los
                       | rlop-lrdi
                       | rlop-lais
                       | rpop-oof
                       | rpop-pls
                       | rpop-lop
                       | rpop-ais
                       | rpop-prdi
                       | rpop-erdi
                       | rxcp-lcd
                       | rxcp-oocd
                       | raop-oof
                       | raop-lof
                       | rase-sf
                       | rase-sd }
```

The alarm names should be interpreted based on the meaning of these acronyms:

- `OOF`: Out of Frame, error counting a misaligned frame
- `LOF`: Loss of Framing, a bit rate is received, but the interface cannot achieve frame delineation.
- `LOS`: Loss of Signal
- `AIS`: Alarm Indication Signal, signal indicating forward that a node along the STM-1/SDH path is defect. It may be preceded by the letter 'P' (Path), 'L' (Line), 'S' (Section); see SDH related

information.

- `RDI`: acknowledgement of the AIS message.
- `RSOP`: Receive Section Overhead Processor
- `RLOP`: Receive Line Overhead Processor
- `RPOP`: Receive Path Overhead Processor
- `RXCP`: Receive ATM Cell Processor (ATM Cell delineation)
- `RASE`: Receive BER monitor
- `OOCD`: Out Of Cell Delineation

Once done with these configuration parameters, you can terminate the STM-1 interface configuration using the following:

```
CLI(stm1)> execute
CLI(stm1)> exit
CLI(config-if)> exit
CLI(configure)>
```

2.3.9.2 Statistics

Use the following command to display STM-1 statistics:

```
CLI> show stm1
```

2.3.9.3 Configuration Example

The following example provisions an STM-1 interface with a single ATM VC.

```
interface atm 0
stm1
no report all
execute
exit
exit
interface atm 0.4
pvc ipoa vpi 0 vci 34
ip address 20.24.4.12 255.255.255.0
ip remote 20.24.4.25
inarp no
qos cbr pcr 64000000
priority 2
execute
exit
exit
```

2.4 ATM GENERIC PERMANENT CONNECTIONS (GENERIC PVC)

Some PVC types only support a dedicated service (IPoA, PPPoA …). ATM-AAL5 PVC's are generic PVC meant for multiple applications (for example: VLAN over ATM or MLPPPoEoA). This chapter deals with PVC for generic applications (generic PVC). Refer to 2.5 for PVC for dedicated services (dedicated PVC).

The generic PVC configuration takes place at the ATM sub-interface level and is well defined by the following parameters: ATM interface, ATM-AAL5 sub-interface upon which the PVC is linked, and VPI and VCI numbers of the PVC.

2.4.1 PVC Templates

A PVC template is intended to configure some common PVC characteristics. To start configuring the PVC template that contains the ATM related parameters, enter:

```
CLI(configure)> virtual-template pvc <1..255>
CLI(config-virt-pvc)>
```

The ATM Quality of Service (QoS) on PVC is configured in this virtual template by defining the Class of Service (CoS) type and its associated rate parameters.

The `pcr` rate parameter represents the peak cell rate in bit per second (bps). The `pr` rate parameter represents the peak rate (=PCR) in bit per second (bps). The `arb` rate parameter is the average bit rate of the stream in bit per second (bps). The `scr` rate parameter is the sustainable cell rate in bit per second (bps). The `mbs` rate parameter represents the maximum burst size in cells (number of cells). The `mcr` rate parameter is the minimum (guaranteed) cell rate in bit per second (bps).

Note: If a `PCR` value is configured above the actual interface speed, this value is automatically reduced to the actual interface speed.

To set the UBR CoS (for all PVC), use the following command in virtual template configuration mode:

```
CLI(config-virt-pvc)> qos ubr pcr <32000..line-rate>
```

To set the CBR CoS (for all PVC), use the following command in virtual template configuration mode:

```
CLI(config-virt-pvc)> qos cbr pcr <32000..line-rate>
```

To set the VBR-rt CoS (for all PVC) use the following command in virtual template configuration mode VBR-rt should be used for voice service.

```
CLI(config-virt-pvc)> qos vbr-rt pr <32000..line-rate>
                                 arb <32000..pr-value>
```

To set the VBR-nrt CoS (for all PVC) use the following command in virtual template configuration mode:

```
CLI(config-virt-pvc)> qos vbr pcr <32000..line-rate>
                          scr <32000..pcr-value> mbs <burst-size>
```

To set the UBR+ CoS (ONE400 specific) use the following command in virtual template configuration mode.

```
CLI(config-virt-pvc)> qos ubrplus pcr <32000..line-rate>
                                   mcr <32000..pcr-value>
```

To reset the default CoS (UBR, 2 Mbps PCR) use the following command in virtual template configuration mode.

```
CLI(config-virt-pvc)> qos qos-default
```

To apply a priority to the CoS, use the following command in virtual template configuration mode.

```
CLI(config-virt-pvc)> priority <1..8>
```

The `priority` command is optional and it is not recommended to use it in standard cases. You will find below a table summarizing ATM CoS for the OneOS-based range of products. The table explains the meaning of the `priority` command and its associated priority values. The priority is the order in which the internal ATM scheduler sends ATM cells. If two PVC's with the same CoS must be differentiated, then the priority is significant.

Class of Service	ONE400 device		ONE20G/D-50-60-70-80-80XM-100D/M-150-200-270-300-700 devices		ONE20A-100A devices	
	Possible Priorities	Default	Possible Priorities	Default	Possible Priorities	Fixed
CBR	1,2	2	1	1	N/A	1
VBR-rt	3,4	4	1,2	2	N/A	N/A
VBR-nrt	5,6	5	3,4	3	N/A	2
UBR+	5,6	6	N/A	N/A	N/A	N/A
UBR	7,8	8	3,4	4	N/A	4

The ATM OAM parameters are set under this template. Please refer to 2.6.2 for more details. Once all parameters in virtual template are set, enter the `exit` command.

2.4.2 Generic PVC Creation

To enter in ATM sub-interface configuration mode, use the following command in configuration mode:

```
CLI(configure)> interface atm-aal5 <port 0..4>.<sub-interface 1..255>
CLI(config-if)>
```

The port number is the one assigned to the ATM interface, while the ATM sub-interface number is an arbitrary value.

To enter into generic PVC configuration mode, use the following command in ATM AAL5 sub-interface configuration mode (if the PVC does not exist, it is created):

```
CLI(config-if)> pvc vpi <vpi> vci <vci> [<vc-name>]
CLI(aal5)>
```

The VPI and VCI values define the PVC. The values are chosen from the range defined in the ATM driver interface configuration mode. `<vc-name>` is a string of characters identifying the PVC connection. The command may return an error if the couple (VPI-VCI) exists for another ATM sub-interface.

2.4.3 Generic PVC Configuration

The generic PVC needs to be configured using a PVC virtual template that gathers all the parameters and has to be defined before (see 2.4.1). To associate the virtual template parameters to the generic PVC being configured, use the following command in AAL5 PVC configuration mode:

```
CLI(aal5)> profile-pvc <virtual-template-id>
```

Then, return to the configuration mode by entering:

```
CLI(aal5)> exit
CLI(config-if)> exit
CLI(configure)>
```

2.4.4 Generic PVC statistic counters reset

Use the following command in global mode to reset the statistics of all generic PVC:

```
CLI> show statistics atm-aal5 reset
```

2.4.5 Generic PVC configuration example

The following example illustrates the configuration of PPP over Ethernet over VLAN over ATM PVC. It creates an ATM PVC (VPI 8, VCI 35) with one VLAN (10) in the PVC. The default gateway uses PPPoE over the VLAN (PPPoEoVLAN).

```
configure terminal
 virtual-template ppp 1
  ipcp dns-accept
  mru local 1492 no
  mru remote 1492 yes
  authentication chap/pap
  username test123
  password test123
  reconnection automatic
  execute
 exit
 bba-group pppoe 3
  virtual-template 1
  service-name ANY
 exit
 interface adsl 0
  execute
  exit
 interface atm 0
  driver ident 0
   max channels 3
   max vp 3
   max vc 3
   range vp min 0 max 10
   range vc min 30 max 120
   execute
  exit
  adsl
   execute
   exit
 exit
 virtual-template pvc 4
  execute
 exit
 interface atm-aal5 0.1
  pvc vpi 8 vci 35
   profile-pvc 4
  exit
 exit
 interface atm-aal5 0.1.10
  encapsulation dot1Q 10
  pppoe enable 3
  pppoe-client dial-pool-number 100
 exit
 ip route 0.0.0.0 0.0.0.0 dialer 1
 interface dialer 1
  encapsulation ppp
  dialer pool 100
 exit
```

2.5 ATM DEDICATED PERMANENT CONNECTIONS (DEDICATED PVC)

The dedicated PVC configuration takes place at the ATM sub-interface level and is well defined by the following parameters: ATM interface, ATM sub-interface upon which the PVC is linked, VPI and VCI numbers of the PVC, and type of PVC.

Other parameters are optional or mandatory, according to the type of PVC created.

2.5.1 Dedicated PVC Attachment

To enter in ATM sub-interface configuration mode, use the following command in configuration mode (if the PVC does not exist, it is created):

```
CLI(configure)> interface atm <port-number 0-4>.<sub-interface 1-255>
```

The port number is the one assigned to the ATM interface, while the ATM sub-interface number is an arbitrary value.

To enter into dedicated PVC configuration mode, use the following command in ATM sub-interface configuration mode:

- For a PVC used by IP:

```
CLI(config-if)> pvc { ipoa | pppoa | pppoeoa } vpi <vpi> vci <vci>
                                                          [<vc-name>]
```

`ipoa` indicates that the PVC is used for IP traffic directly encapsulated in AAL-5 (no PPP); whereas `pppoa` indicates that the PVC supports PPP for IP traffic. To remove the ATM sub-interface configuration, use the following command in configuration mode.

```
CLI(configure)> no pvc { ipoa | pppoa | pppoeoa } vpi <vpi> vci <vci>
```

- For a PVC used by voice over ATM (VoDSL) service:

```
CLI(config-if)> pvc voiceoa vpi <vpi> vci <vci> [<vc-name>]
```

The VPI and VCI values define the PVC. The values are chosen from the range defined in the ATM driver interface configuration mode. `<vc-name>` is a string of characters identifying the PVC connection. The command may return an error if the couple (VPI-VCI) exists for another ATM sub-interface.

2.5.2 Dedicated PVC Configuration

Note: in the following `CLI(pvc)>` is used for any dedicated PVC (`ipoa`, `pppoa`, `pppoeoa` and `voiceoa`).

To set PVC encapsulation of IP packets within AAL5 frames, use the following command in PVC configuration mode:

```
CLI(pvc)> encapsulation { llc | mux }
```

Encapsulation is set to `llc` (LLC/SNAP) by default. The reader should refer to RFC 2684 for details about IP encapsulation in AAL-5.

The ATM Quality of Service (QoS) on PVC is configured in PVC configuration mode by defining the Class of Service (CoS) type and its associated rate parameters. Refer to 2.4.1 for more information about the QoS configuration commands.

```
CLI(pvc)> qos { ubr pcr <pcr-in-bps> |
                cbr pcr <pcr-in-bps> |
                vbr-rt pr <pr-in-bps> arb <arb-in-bps> |
                vbr pcr <pcr-in-bps> scr <scr-in-bps> mbs <mbs-in-cells>|
                        ubrplus pcr <pcr-in-bps> mcr <mcr-in-bps> }
```

2.5.2.1 IP over ATM Service

IPOA service is well identified by a PVC and an IP address, defining the PVC as an IP interface. To enter in IPOA configuration mode, use the following command in ATM sub-interface configuration mode:

```
CLI(config-if)> pvc ipoa vpi <vpi> vci <vci>
```

To set the IP address for an IPOA PVC, use the following command in IPOA configuration mode:

```
CLI(ipoa)> ip address <ip address> <subnet mask>
```

The Inverse Address Resolution Protocol (INARP) is used by default for resolution of the IP address of the remote endpoint of the PVC. If you want to provide the IP address of the remote end point of the PVC, the INARP must be disabled. The following command disables the ATM Inverse ARP and sets the remote IP address:

```
CLI(ipoa)> inarp no

CLI(ipoa)> ip remote <ip-address>
```

To enable the Inverse ARP for ATM protocol on an IPOA PVC, use the following command in IPOA configuration mode:

```
CLI(ipoa)> inarp client

CLI(ipoa)> execute
```

Configuration steps are as follows:

```
CLI> configure terminal
CLI(configure)> interface atm <index1>.<index2>
CLI(config-if)> pvc ipoa vpi <number1> vci <number2>
CLI(config-if)> encapsulation { llc | mux }
CLI(ipoa)> qos <type> pcr <value> scr <value> mbs <value>
CLI(ipoa)> execute
```

The encapsulation default value is LLC.

Example:

```
interface atm 0.1
pvc ipoa vpi 0 vci 32
qos ubr pcr 2000000
ip address 20.1.0.102 255.255.255.0
execute
exit
exit
```

By default, an IPoA interface is of type BROADCAST-MULTICAST. The fact that it is a BROADCAST interface means that the sub-network associated with the IP address/mask contains a network address (lowest address in IP subnet) and a broadcast address (highest address). The IP address of the PVC must be different from network and broadcast address. For example, if IP address is 10.10.10.1/30, the network address is 10.10.10.0 and broadcast address is 10.10.10.3. If you are using a /31 subnet mask, only two addresses are available (network and broadcast). To use those IP as interface IP, please enter the command `no ip directed-broadcast` that removes the BROADCAST behavior of the interface. Example:

```
interface atm 0.1
  pvc ipoa vpi 0 vci 32
  qos ubr pcr 2000000
  ip address 20.1.0.102 255.255.255.254
  execute
 exit
 no ip directed-broadcast
exit
```

2.5.2.2 PPP over ATM Encapsulation

2.5.2.2.1 Configuration

To enter in PPPoA configuration mode, use the following command in ATM sub-interface configuration mode:

```
CLI(config-if)> pvc pppoa vpi <vpi> vci <vci>
```

PPP provides a standard method of encapsulation across a Point-to-Point connection. Different parameters may be negotiated for this connection.

2.5.2.2.2 IP Address

The IP address must be set if the IPCP dynamic is not used. By default, IPCP dynamic is set and the IP address is set to 0.0.0.0. To enter the IP address, use the following command:

```
CLI(pppoa)> ip address <ip-address> [<mask>]
```

Instead of having an IP address, the local PPP interface can be unnumbered:

```
CLI(pppoa)> ip unnumbered <interface-type> <unit>
```

2.5.2.2.3 Encapsulation Type

This command specifies the encapsulation type used for this PVC.

```
CLI(pppoa)> encapsulation { llc | mux }
```

2.5.2.2.4 IPCP Parameters

If the command `no ipcp` is entered, this feature is disabled and the IP address must then be configured.

If `static` mode is selected, the IPCP service functions in static mode. In this case the IP address must be configured and is checked by the remote equipment (the OneOS-based router and the remote equipment must have the same data configured -such as the IP address-).

If `dynamic` mode is selected, the IP address for the interface is obtained via PPP/IPCP from the next hop router.

```
CLI(pppoa)> ipcp { static | dynamic }
```

If `dynamic` mode is selected, the network mask can also be obtained via PPP/IPCP.

```
CLI(pppoa)> ipcp mask-request
```

Note: If `mask-request` is used, the "`ip unnumbered <interface>`" command is mandatory for the IP address and the network mask apply to `<interface>` otherwise they will apply to PPP.

If you want to retrieve the DNS server address per IPCP use the next command:

```
CLI(pppoa)> ipcp dns-accept
```

2.5.2.2.5 MRU Parameters

The MRU is set using a command with the following format:

```
CLI(pppoa)> mru local <size> { yes | no } { yes | no }
CLI(pppoa)> mru remote <size> { yes | no } { yes | no }
```

This command sets the Maximum Receive Unit (MRU) local/remote length and enables selection of negotiation mode. The MRU designates the maximum size of packets received in a PPP payload.

If negotiation request is enabled (`yes` in first yes-no argument), the MRU length is negotiated with the remote PPP peer and is lower or equal to the `size` value. The negotiation accept (second yes-no argument) defines whether the MRU proposed by the remote peer should be accepted. If negotiation is not enabled (`no` in first or second yes-no argument), the maximum datagram length is equal to `size` value. At any rate, the length of MRU is between 40 and 9180 bytes (default value 1500 bytes).

2.5.2.2.6 Authentication Type

Use this command to specify the authentication type.

```
CLI(pppoa)> authentication { no | pap | chap | chap/pap }
                                   [two-ways | one-way-callin | one-way-called]
```

In case of PAP authentication, the CPE must send a name and password that is checked against a matched entry in the local database of the remote router.

In case of CHAP authentication, the CPE sends a "challenge" message to the remote device, which then encrypts the value with a shared key and returns it to the CPE.

To disable this function: use the `authentication no` command. By default, `chap` is enabled.

The second argument defines the authentication types. They are:

- `two-ways`: authentication is done by both sides.
- `one-way-callin` (default): authentication material is sent to the remote device, but the remote device is not locally authenticated.
- `one-way-called`: authentication material is not sent to the remote device (only username is sent), but the remote device must be authenticated.

Note: for the current release, only bi-directional CHAP and CHAP server are supported. All other combinations are supported with the attribute `one-way-callin`.

2.5.2.2.7 Username and Password

This configuration command specifies the username and password (character string) used by the authentication protocol.

The username may content a predefined string of the form **#mac**<i># that will be replaced during authentication by the corresponding MAC address of the device.

Example: `#mac1#@my.com` will be replaced by `0012EF44E276@my.com` where `00:12:EF:44:E2:76` is the MAC address #1 of the device as listed in the product information area.

The maximum permitted length of username is 64 characters; the maximum permitted length of password is 30 characters when entered in clear text and 128 characters when entered already encrypted.

```
CLI(pppoa)> username <username>
CLI(pppoa)> password <password> [<crypto-algorithm>]
```

If `crypto-algorithm` is not provided, the password is entered and stored in clear text.
If `crypto-algorithm = 0` is provided, the password is entered in clear text and stored encrypted.
If `crypto-algorithm = 1` is provided, the password is entered and stored encrypted.

2.5.2.2.8 Magic Number Negotiation

Magic number is used by the LCP protocol to detect looped-back lines and to check the continuity using "echo" frames.

The magic number may be negotiable: the remote device can choose the number; otherwise (if non-negotiable) the CPE selects its own number.

```
CLI(pppoa)> magic { negotiation | not-negotiable }
```

2.5.2.2.9 Keep-Alive Timer

The keepalive mechanism is enabled by default. It sends periodical "echo" frames to prevent the disconnection of the line after a long period of PPP inactivity. However, after a certain number of unsuccessful retries of "echo" frames sending the line is then disconnected. Use the following command to modify the period timer (4 seconds by default) and optionally the number of retry (20 retries by default).

```
CLI(pppoa)> [no] keepalive <timer:0..3600> [retry <1..100>]
```

The no form of the command disables the keepalive mechanism. Note that the keepalive mechanism is also disabled when the timer is set to 0.

2.5.2.2.10 Retry Timer

This timer is used by the LCP protocol during the configuration and negotiation phase for connection and auto-reconnection. If a BAS is down and up again all routers will attempt to connect to the BAS simultaneously, thus flooding the BAS with connection requests. A random timer for session re-establishment can prevent several routers from reconnecting at the same time. The configuration must provide a minimum and a maximum range for retry timers. The value of the timer is then chosen randomly between the values min and max.

```
CLI(pppoa)> retry-time min <seconds> max <seconds>
```

Reconnection (default value is `no reconnection`):

```
CLI(pppoa)> [no] reconnection
```

This command is used, after session shutdown, to enable session reconnection.

2.5.2.2.11 Link Fragmentation and Interleaving (LFI)

LFI enables to reduce significantly transit delays for priority packets. To understand the under-laying properties of the IP QoS, please refer to "Transit Delay Control" section in "Quality of Service" chapter of "OneOS – Basic IP User Guide" document. When activating LFI, the priority packets are encapsulated in PPP and low priority packets are transported in MLPPP. If low priority packets are too large, they are fragmented. Priority packets are transmitted after the current frame being emitted by the interface.

To enable interleaving, use the following command:

```
CLI(pppoa)> multilink interleave
```

Then you must set some MLPPP parameters:

```
CLI(pppoa)> multilink discriminator <class> [<string>]

CLI(pppoa)> multilink { fragment-size <bytes> | fragment-delay <ms> }

CLI(pppoa)> multilink header { long | short }

CLI(pppoa)> multilink max-rcvunit <bytes>
```

The discriminator is used to identify to which bundle a link must be added. It must be usually configured (it depends on the remote end of the MLPPP connection). `class` is an integer ranging from 1 to 5 and `string` is a string of 20 characters maximum. If the class value is 3 (IEEE 802.1 Globally Assigned MAC Address) with string, OneOS uses the string as discriminator to open the MLPPP connection. If the class value is 3 (IEEE 802.1 Globally Assigned MAC Address) without string, then OneOS uses the MAC address of FastEthernet 0/0 to open the MLPPP connection. If the `class` parameter is other than 3, the `string` parameter is mandatory.

The maximum fragment size is either determined by the `multilink fragment-size <bytes>` command or, it is calculated from PVC line rate divided by the fragment delay. If `fragment-delay` is configured, the fragment size is re-calculated whenever the DSL line is resynchronized.

The `max-rcvunit` is the Maximum Remote Receive Unit (MRRU) and is by default 1500 bytes.

2.5.2.2.12 PPPoA session re-initialization

The PPPoA session can be automatically re-initialized (disconnected then restarted) using the following command in PPPoA configuration mode:

```
CLI(pppoa)> reinit <down-time> <mid-time> [<up-time>] [<day-of-the-week>]
```

The session will be disconnected then restarted every day at a random time between down and mid times. Use the optional `day-of-the-week` parameter to have it disconnected then restarted once a week at the defined day (Monday to Sunday).

Use the optional `up-time` parameter to have the session restarted at a random time between mid and up times (i.e. down for a randomly duration of time).

To restore default settings and not use automatic re-initialization of the PPPoA session, use the no form of the above command:

```
CLI(pppoa)> no reinit
```

2.5.2.2.13 Example

```
interface atm 0.1
 pvc pppoa vpi 0 vci 34
  encapsulation mux
  ipcp dynamic
  authentication pap
  username user2_0
  password oneaccess
  mru local 1500 no
  mru remote 1500 no
  magic number not-negotiable
  qos ubr pcr 500000
  execute
  exit
 exit
```

2.5.2.2.14 Statistics

In case of PPPoA PVC, a statistics command is available. It permits the display of different information regarding the steps required for PPP connection.

```
CLI> show statistics pvc pppoa 0

PPPoA PVC Statistics of ATM interface 0

PVC Pppoa Statistics: vcd = 1, vpi = 0, vci = 33, vcn =

LCP Statistics:
PPP LCP: STOP
Local MRU: 1500, Remote MRU: 1500
tx packets: 88, tx bytes: 168
rx packets: 58, rx bytes: 596
packets rx > max size: 0
packets rx < min size: 0
tx packets Config Req: 10, rx packets Config Req: 1
tx packets Config Acq: 1, rx packets Config Acq: 1
tx packets Config Nak: 0, rx packets Config Nak: 1
tx packets Config Rej: 0, rx packets Config Rej: 0
tx packets Termination Req: 2, rx packets Termination Req: 0
tx packets Termination Acq: 0, rx packets Termination Acq: 0
tx packets Code Rej: 0, rx packets Code Rej: 0
tx packets Protocol Rej: 0, rx packets Protocol Rej: 0
tx packets Echo Req: 72, rx packets Echo Req: 3
tx packets Echo Rep: 3, rx packets Echo Rep: 52
tx packets Discard Req: 0, rx packets Discard Req: 0
tx packets Identification: 0, rx packets Identification: 0
tx packets Time remaining: 0, rx packets Time remaining: 0

NCP Statistics:
PPP IPCP: STOP
tx packets: 2, tx bytes: 36
rx packets: 2, rx bytes: 36
Time for last establishing PPP connection 42.00 seconds
Time for last established PPP connection 73.00 seconds
Time since last stopped PPP connection 2024.30 seconds
```

2.5.2.3 PPP over Ethernet over ATM Encapsulation (PPPoEoA)

PPPoEoA is a WAN protocol, which enables the transport of PPPoE frames bridged in LLC/AAL5 PDU. PPPoEoA is traditionally used over DSL uplinks. Indeed, there used to be DSL modems offering the customer an Ethernet interface to DSL access. The DSL modem functions as an Ethernet bridge. To allow authentication of devices sending traffic through the DSL modem, the PPPoE protocol, running on a PC, is required so that the BAS can grant an access to the IP network after PAP/CHAP authentication.

As the ONE range of products can have an integrated DSL modem, the PPPoE protocol as well as the bridging function is managed by OneOS to allow compatibility with BAS not supporting this protocol.

PPPoE is actually made up of three sub-components:

- The PPP protocol, which has the same properties and configuration as the PPPoA protocol
- The PPPoE protocol, defined in RFC 2516, which manages the connection between a client (a host)

and a PPPoE service concentrator

- The bridging: PPPoE frames are bridged in AAL-5 PDU using the LLC-SNAP encapsulation

A BAS is referenced as PPPoE service concentrator in the PPPoE terminology.

2.5.2.3.1 Configuration

To enter in PPPoEoA configuration mode, use the following command in ATM sub-interface configuration mode:

```
CLI(config-if)> pvc pppoeoa vpi <vpi> vci <vci>
CLI(pppoeoa)>
```

The user must then enter PPPoE specific parameters. The first parameter is the source MAC address, which is used by the device to generate the PPPoE frames:

```
CLI(pppoeoa)> ethernet address <AA:BB:CC:DD:EE:FF>
```

Default: MAC address of the LAN port.

PPPoE frames are normally emitted on a LAN as broadcast frames during PPPoE initiation. Several PPPoE service concentrators may be found on the LAN. Therefore, names must identify the clients and servers. The following commands identify the ONE device (service name) and the concentrator name:

```
CLI(pppoeoa)> service name <string>
CLI(pppoeoa)> concentrator name <string>
```

By default, the PPPoEoA max MRU is 1492 bytes as per RFC 2516. Use the following command to allow PPPoEoA MTU/MRU greater than 1492 bytes according to RFC 4638. In that case, the value used is the MRU value defined for PPP. Use the `no` form of the command to not handling MRU greater than 1492 bytes for PPPoEoA.

```
CLI(pppoeoa)> tag ppp-max-payload
```

When a customer uses both PPPoEoA and dialer PPPoE for backup with the same PPP account username/password, the following may happen: when ATM goes back UP state after being DOWN state and dialer PPPoE connected, it happens the PPPoEoA does not succeed to go back UP because authentication fails. The following command shuts down the specified dialer interface when the PPPoEoA PVC under which that command is configured receives a PADT (PPPoE Active Discovery Terminate). The dialer specified goes back to normal state (i.e.: no shutdown state) 90 seconds after the PPPoEoA PADT reception. This command can be entered several times in case more that one dialer PPPoE is used for backup.

```
CLI(pppoeoa)> padt clear-pppoe-dialer <id>
```

Then, the PPP parameters can then be entered. They are identical to those of PPPoA (see 2.5.2.2). Once done with the PPP parameters, do not forget to enter the `execute` command before leaving the PPPoEoA interface configuration mode.

2.5.2.3.2 Example

The following configuration provides an example for PPPoEoA configuration:

```
interface virtual-template 2
ip nat inside overload
exit

interface atm 0.3
! PPPoE allows 1492 byte long IP payloads
! The following commands clamp TCP MSS fields with a shorter MSS
! than the Ethernet should allow.
ip tcp adjust-mss 1452
pvc pppoeoa vpi 0 vci 40
ipcp static
ip address 20.12.3.4
mru local 1492 yes
mru remote 1492 no
```

```
authentication chap
username ONEACCESS
password ******
concentrator name concentrator
keepalive 5
retry-time min 3 max 5
source 2
execute
exit
exit
```

2.5.2.3.3 Statistics

To display PPPoEoA statistics, use the following command:

```
CLI> show statistics pvc pppoeoa <atm-if>
```

```
PPPoEoA PVC Statistics of ATM interface 0

PVC Pppoeoa Statistics: vcd = 3, vpi = 0, vci = 40, vcn =

LCP Statistics:
PPP LCP: OPEN
Local MRU: 1492, Remote MRU: 1492
tx packets: 4, tx bytes: 32
rx packets: 4, rx bytes: 56
packets rx > max size: 0
packets rx < min size: 0
tx packets Config Req: 2, rx packets Config Req: 1
tx packets Config Acq: 1, rx packets Config Acq: 1
tx packets Config Nak: 0, rx packets Config Nak: 1
tx packets Config Rej: 0, rx packets Config Rej: 0
tx packets Termination Req: 0, rx packets Termination Req: 0
tx packets Termination Acq: 0, rx packets Termination Acq: 0
tx packets Code Rej: 0, rx packets Code Rej: 0
tx packets Protocol Rej: 0, rx packets Protocol Rej: 0
tx packets Echo Req: 0, rx packets Echo Req: 1
tx packets Echo Rep: 1, rx packets Echo Rep: 0
tx packets Discard Req: 0, rx packets Discard Req: 0
tx packets Identification: 0, rx packets Identification: 0
tx packets Time remaining: 0, rx packets Time remaining: 0

NCP Statistics:
PPP IPCP: OPEN
local address is 20.12.3.4 remote Addr is 20.12.3.3
tx packets: 3, tx bytes: 72
rx packets: 3, rx bytes: 48
packets rx > max size: 0
packets rx < min size: 0
Time for establishing PPP connection 2.18 seconds
Time since established PPP connection 11.62 seconds

DISCOVERY Statistics:
DISCOVERY State: ESTABLISH SESSION
local address is 1a:2b:3c:4d:5e:70 remote Addr is 00:02:b9:38:19:90

OEOA Statistics:
total tx packets: 7, tx bytes: 144
total rx packets: 13, rx bytes: 1808
```

To display PPPoEoA interface configuration details as well as its default value, use the '`show configuration`' command:

```
CLI(configure)> show configuration pvc pppoeoa <atm-if>
```

```
List of configured PPPoEoA PVC
=================================================================
Configuration of PVC pppoeoa: sub-int = 3, vpi = 0, vci = 40, vcn =
=================================================================
ATM interface index: 0
sub-interface: 3
vpi number: 0
vci number: 40
vcn:
pvc Encapsulation: aal5/llc
QOS: UBR
Peak Cell Rate: 2000000.00
Priority: 4
IP address for this int: 20.12.3.4
Ipcp: Static negociation
```

```
Local MRU: 1400
Local MRU negociation: Yes
Remote MRU: 1400
Remote MRU negociation: No
Authentication: PAP
User name: ONEACCESS
password: Yoohoo
Use magic number: No
pvc keep alive: 5
pvc retry timer: min = 3 max = 5
MAC address: 1a:2b:3c:4d:5e:70
Concentrator name: weather
Service name: meteo
pvc reconnection: no

OAM Pvc level:
ais_rdi_enable: 0
lb_ete_enable: 0
lb_seg_enable: 0
cc_seg_enable: 0
cc_ete_enable: 0
lb_retries_down_to_up: 3
lb_retries_up_to_down: 5
lb_retry_frequency: 1
counter_aisrdi_up_to_down: 3
timer_aisrdi_down_to_up: 3
number_retries: 0
manage: 1
intrusive: 1
lb_normal_frequency: 10
```

To debug connection issues with PPPoE, use the `event` facility:

```
CLI> event filter add wan pppoeoa all show
```

2.5.3 Multi-Link PPP (MLPPP) over ATM and MLPPPoEoA

2.5.3.1 Configuration

MLPPP over ATM is only supported on the ONE80/180/300 routers in V3.7R15. It makes it possible to inverse-multiplex an IP stream over multiple SHDSL lines. If the inverse multiplex function of MLPPP is not required, but only the interleaving (LFI) function of MLPPP, please refer to 2.5.2.2.11.

MLPPP is configured as follows:

- Virtual ATM template: template of ATM parameter configuration (ATM shaping, OAM, …)
- PVC configuration: the VPI/VCI is configured and attached to a virtual ATM template. It is also attached to a MLPPP dialer interface
- A MLPPP dialer-interface: this is the logical IP interface running the MLPPP protocol. The IP services are attached to this interface (access list, QoS …). The dialer interface is attached to a PPP virtual template containing the PPP/MLPPP specific protocol parameters.

The PPP parameters must be set for authentication and PPP parameters. All PPP parameters are entered within a PPP virtual template. This PPP virtual template must be set to automatic reconnection. Please refer to 5 for more details. Example:

```
virtual-template ppp 1
 ipcp dynamic
 authentication chap
 ipcp dns-accept
 username apassword@provider.com
 password ******
 reconnection automatic
 execute
exit
```

The MLPPP dialer interface is attached to a pool number. The pool number is an identifier shared by the physical interface and the dialer interface. To configure an MLPPP dialer interface, attach a PPP virtual template and a pool number (when these two objects are attached, and only after, IP configuration can be set):

```
CLI(configure)> [no] interface dialer-mlppp <0-65535>
CLI(config-if)> pool-number <1-255>
CLI(config-if)> profile <0-10>
CLI(config-if)> exit
```

By default, the protocol is MLPPPoA; extra configuration parameters are required for MLPPPoEoA:

- Mode: PPPoE client (host) or server (concentrator). A CPE router is usually in host mode
- Service name: the service name is a string transmitted during PPPoE establishment; this string is usually not checked by the network. The default service name is ‘any service’.
- Concentrator name (default: empty)

For an MLPPPoEoA client, the configuration commands are:

```
CLI(configure)> interface dialer-mlppp <0-65535>
CLI(config-if)> protocol pppoeoa
CLI(config-proto)> service name <name>
CLI(config-proto)> no service name
CLI(config-proto)> exit
CLI(config-if)> exit
```

For an MLPPPoEoA server:

```
CLI(configure)> interface dialer-mlppp <0-65535>
CLI(config-if)> protocol pppoeoa
CLI(config-proto)> mode concentrator
CLI(config-proto)> concentrator name <name>
CLI(config-proto)> no concentrator name
CLI(config-proto)> service name <name>
CLI(config-proto)> no service name
CLI(config-proto)> exit
CLI(config-if)> exit
```

Then, attach the dialer interface to ATM PVC and re-use the same pool-number:

```
CLI(configure)> interface atm-aal5 <0-3>.<1-255>
CLI(config-if)> mlppp dial-pool <1-255>
CLI(config-if)> pvc vpi <0-255> vci <0-2048>
CLI(config-pvc)> profile-pvc <virtual-template-id>
CLI(config-pvc)> exit
CLI(config-if)> exit
```

For more information on ATM-AAL5 PVC, please refer to section 2.4.

2.5.3.2 Example

MLPPPoA:

```
interface atm 0
 gshdsl 0
  execute
 exit
exit
interface atm 1
 gshdsl 1
  execute
 exit
exit
interface atm 2
 gshdsl 2
  execute
 exit
exit
interface atm 3
 gshdsl 3
  execute
 exit
exit
virtual-template pvc 1
```

```
 qos cbr pcr 2304000
exit
virtual-template ppp 1
 ipcp dynamic
 authentication chap
 ipcp dns-accept
 username apassword@provider.com
 password ******
 reconnection automatic
 execute
exit
interface dialer-mlppp 0
 pool-number 0
 profile ppp 1
 ip nat inside overload
 service-policy output shaping
exit
interface atm-aal5 0.1
 mlppp dial-pool 0
 pvc vpi 1 vci 32
  profile-pvc 1
 exit
exit
interface atm-aal5 1.1
 mlppp dial-pool 0
 pvc vpi 1 vci 32
  profile-pvc 1
 exit
exit
interface atm-aal5 2.1
 mlppp dial-pool 0
 pvc vpi 1 vci 32
  profile-pvc 1
 exit
exit
interface atm-aal5 3.1
 mlppp dial-pool 0
 pvc vpi 1 vci 32
  profile-pvc 1
 exit
exit
```

MLPPPoEoA:

```
interface atm 0
 gshdsl 0
  execute
 exit
exit
interface atm 1
 gshdsl 1
  execute
 exit
exit
interface atm 2
 gshdsl 2
  execute
 exit
exit
interface atm 3
 gshdsl 3
  execute
 exit
exit
virtual-template pvc 1
 qos cbr pcr 2304000
exit
virtual-template ppp 1
 ipcp dynamic
 authentication chap
 ipcp dns-accept
 username apassword@provider.com
 password ******
 reconnection automatic
 execute
exit
interface dialer-mlppp 0
 pool-number 0
 profile ppp 1
 protocol pppoeoa
 exit
 ip nat inside overload
 service-policy output shaping
exit
interface atm-aal5 0.1
 mlppp dial-pool 0
 pvc vpi 1 vci 32
  profile-pvc 1
 exit
exit
interface atm-aal5 1.1
 mlppp dial-pool 0
 pvc vpi 1 vci 32
  profile-pvc 1
 exit
exit
interface atm-aal5 2.1
 mlppp dial-pool 0
 pvc vpi 1 vci 32
  profile-pvc 1
 exit
exit
interface atm-aal5 3.1
 mlppp dial-pool 0
 pvc vpi 1 vci 32
  profile-pvc 1
 exit
exit
```

2.5.3.3 Debug and Statistics

To display the MLPPP configuration information:

```
CLI> show configuration interface dialer-mlppp <id>
```

To display the MLPPP dialer statistics:

```
CLI> show statistics dialer-mlppp <id>
```

To debug the MLPPP protocol, use the next command:

```
CLI> [no] debug dialer-mlppp
```

Use the following command to display statistics regarding the ATM PVC under the sub-interface specified by its number.

```
CLI> show interfaces dialer-mlppp <interface-id>
```

2.6 ATM MANAGEMENT FLOWS

This section describes the main features of the ATM management flow.

2.6.1 Features

2.6.1.1 Definitions

The following terms are used throughout this section:

2.6.1.1.1 OAM

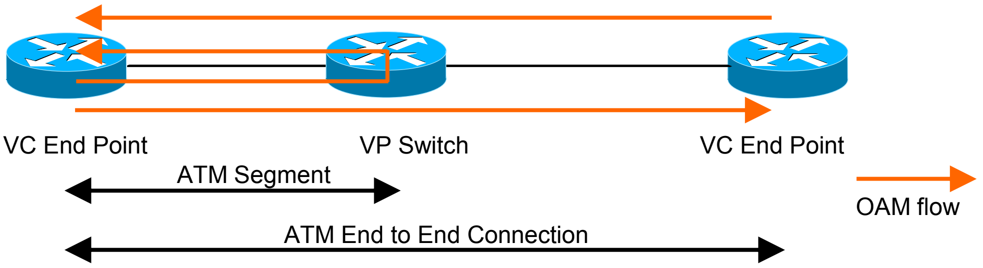
This acronym defines operations and maintenance on broadband networks. This document is specifically related to OAM on ATM networks.

2.6.1.1.2 OAM Flow

In ATM layers, OAM flow stands for the specific ATM cells used for operations and maintenance and that transit between the ATM nodes of an ATM connection. Depending on the cell content, OAM cells are applied to Virtual Paths, or Virtual Channels.

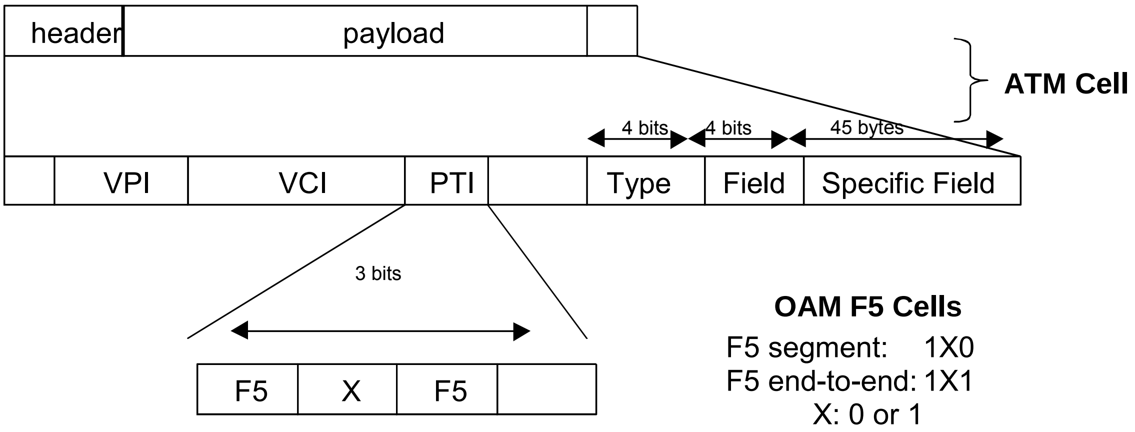
2.6.1.1.3 Description of OAM Information Flow

The OneOS-based router can simultaneously monitor a sub-part or an entire VC connection, thus facilitating interoperability through different operators if needed. The device manages OAM segment and OAM end-to-end cells.



2.6.1.1.4 OAM Cell Structure

The OneOS-based router manages all OAM fault management cells (type set to 0010) and is ITU-T I.610 compliant. The Field of OAM cell structure is coded for the specific cell: AIS, RDI, CC and LB.



The PTI field indicates the type of OAM cell associated with the VP Identifier/VC Identifier. The cell payload depends on the OAM cell type.

2.6.1.2 Handling of ATM OAM

2.6.1.2.1 ATM OAM Prioritization

OAM cells are not processed in the same queue as data. The OAM cells are sent directly without delay over the PVC. As far as the OAM cell shaping and prioritization technique is concerned, the following objectives are fulfilled:

- OAM cells + data traffic must be conform to ATM shaping (i.e. should not exceed PCR, while OAM cells are sent during high traffic load)
- OAM cells have got the highest priority.

2.6.1.2.2 Continuity Check (CC)

In the absence of user flow, CC is a type of cell that can be generated each second by the OneOS-based router on a VC. Because of the periodic reception of cells, the remote ATM end point is ensured that the VC is in operation on the downstream. If the OneOS-based router is configured to receive CC cells, the non-reception of cells for more than 3.5 seconds is interpreted as a VC fault (Loss of Cells, LOC), thus triggering an alarm on the OneOS-based router for the VC. Then, the first received cell triggers a new alarm clearing the VC fault.

The OneOS-based router can be configured to emit and receive either segment CC or end-to-end CC on each VC.

2.6.1.2.3 Loopback (LB)

LB is a type of cell used either periodically or occasionally for testing an ATM segment or a whole ATM connection. The cell is generated with a tag identifier that is kept in memory by the OneOS-based router. The cell is sent and looped at either the next segment or the remote end point of the VC. The reception of a loopback cell with the same tag identifier results in a successful test. Failing loopback tests result from non-reception of loopback cells or reception of cells after the configured timeout.

The OneOS-based router periodically generates LB cells. If LB tests have been successful and are then followed by a failed LB test, the LB sending frequency is changed to generate a retry at a new frequency. A new sample of loopback tests is then carried out at the new frequency. If all the tests fail, the OneOS-based router sets an alarm for the affected VC and the LB sending frequency is changed to normal frequency. As soon as one LB test is successful, a new set of LB tests is carried out. If the series is successful, then an alarm is cleared by the OneOS-based router for this VC.

The OneOS-based router occasionally generates LB cells by means of the ping command. Provided that LB cells are not configured on the VC, a configurable set of loopback cells is sent. The command will provide an estimated rate of successful tests, as well as the average round trip delay of loopback tests.

The OneOS-based router can be configured to send either segment or end-to end LB cells on each VC. When OAM is activated on a VC, all incoming loopback cells on the VC are looped by default.

2.6.1.2.4 Alarm Indication Signal (AIS) and Remote Defect Information (RDI)

AIS and RDI are types of cells used respectively for forwarding an alarm for a VC failure, or a sub-layer default, or acknowledging an AIS alarm or a failure on a VC, or a sub-layer default.

Failures are detected by the OneOS-based router with a LOC alarm or LB alarm at the VC level, or by a physical failure such as Loss of Frames on the DSL interface. As long as a failure remains, RDI cells are generated every second, indicating that a failure has been detected by the end-point.

Alarms Cells are similarly generated. When an AIS cell is received by the OneOS-based router on a given VC, an RDI cell is generated in the reverse direction with the same payload as the AIS cell received. This indicates to the remote end-point that an AIS alarm was received. The RDI and AIS payload contain a 20-byte field containing the address of the ATM node that generated the AIS node (that first detected the failure).

Inversely, when an RDI cell is received by the OneOS-based router on a specific VC, cells are not generated.

The OneOS-based router can be configured to manage AIS and RDI cells on each VC. AIS reception triggers RDI generation. An alarm detected on a VC automatically enables RDI generation.

2.6.2 Permanent Virtual Channel Connectivity

2.6.2.1 *Loopback Ping*

Provided that a PVC manages ATM OAM cells, to generate a set of loopback cells on a PVC, use the following command in global configuration mode:

```
CLI> ping atm port <0-3> vpi <0-255> vci <32-2048>
       { end-loopback | seg-loopback } [repeat <1-255>] [timeout <1-255>]
```

2.6.2.2 *Permanent Virtual Channel Configuration (PVC/SVC)*

OAM is configurable, on each of the above-referenced functionality sets: CC, LB, AIS, and RDI.

2.6.2.3 *Management of a Virtual Channel*

To manage F5 OAM cells and loopback cells received on dedicated PVC, use the following command in PVC configuration mode. Note that management of received F5 OAM is enabled by default.

Note: in the following `CLI(pvc)>` is used for any dedicated PVC (`ipoa`, `pppoa`, `pppoeoa` and `voiceoa`).

```
CLI(pvc)> oam-pvc manage
```

When received OAM cells are not taken into account on the PVC, use the following command:

```
CLI(pvc)> no oam-pvc manage
```

2.6.2.4 *OAM F5 Loopback End-to-End*

To enable F5 OAM end-to-end loopback cell generation and monitoring, use the following commands in VC configuration mode (management of received F5 OAM must be enabled, see 2.6.2.3):

```
CLI(pvc)> oam end-loopback
```

Some parameters can be adjusted:

```
CLI(pvc)> oam-lb retry <up-count> <down-count> <retry-frequency>
CLI(pvc)> oam-pvc [manage] <frequency>
```

`up-count` is the required number of loopback cells to receive, so that the ATM VC can be considered as up (default: 3). Similarly, the `down-count` is the number of lost loopback cells (not looped) required to consider the interface as down (default: 5). When the ATM VC encounters a state change (going up or down), loopback cells are sent temporarily much faster than usual. The `retry-frequency` provides the interval in seconds between each loopback cell emission during this transient phase (default: 1 sec). The `frequency` parameter provides the emission interval of loopback cells for normal working conditions (default: 10 sec).

To disable F5 OAM loopback end-to-end, use the following command:

```
CLI(pvc)> no oam end-loopback
```

F5 OAM end-to-end loopback generation is disabled by default.

2.6.2.5 F5 OAM Loopback Segment

To enable F5 OAM segment loopback cell generation and monitoring, use the following commands in VC configuration mode (management of received F5 OAM must be enabled, see 2.6.2.3):

```
CLI(pvc)> oam seg-loopback
```

Some parameters can be adjusted:

```
CLI(pvc)> oam-lb retry <up-count> <down-count> <retry-frequency>
CLI(pvc)> oam-pvc [manage] <frequency>
```

The parameters are the same as those for F5 OAM loopback end-to-end. To disable F5 OAM loopback segment, use the following command:

```
CLI(pvc)> no oam seg-loopback
```

F5 OAM segment loopback generation is disabled by default.

2.6.2.6 F5 OAM Continuity Check End-to-End

To enable F5 OAM end-to-end continuity-check cells generation and monitoring, use the following commands in PVC configuration mode (management of received F5 OAM must be enabled, see 2.6.2.3):

```
CLI(pvc)> oam end-cc
```

To disable the F5 OAM end-to-end continuity check, use the following command:

```
CLI(pvc)> no oam end-cc
```

The OAM end-to-end continuity check generation is disabled by default.

2.6.2.7 F5 OAM Continuity Check Segment

To enable F5 OAM segment continuity check cells generation and monitoring, use the following commands in PVC configuration mode (management of received F5 OAM must be enabled, see 2.6.2.3):

```
CLI(pvc)> oam seg-cc
```

To disable the F5 OAM segment continuity check, use the following command:

```
CLI(pvc)> no oam seg-cc
```

F5 OAM segment continuity check generation is disabled by default.

2.6.2.8 F5 OAM AIS RDI

To enable F5 OAM AIS and RDI generation and monitoring, use the following commands in PVC configuration mode (management of received F5 OAM must be enabled, see 2.6.2.3):

```
CLI(pvc)> oam ais-rdi
```

To disable F5 OAM AIS RDI management, use the following command:

```
CLI(pvc)> no oam ais-rdi
```

F5 OAM AIS RDI generation is enabled by default.

Some parameters can be optionally adjusted:

```
CLI(pvc)> oam-man aisrdi-down <3-60> aisrdi-up <3-60>
```

The first parameter provides the number of AIS/RDI to consider the interface down, the second to consider the interface up.

2.6.2.9 F5 OAM Intrusive Management

To take into account OAM alarms for changing the availability of ATM PVC, use the following command in PVC configuration mode (management of received F5 OAM must be enabled, see 2.6.2.3):

```
CLI(pvc)> oam-pvc [manage] intrusive
```

The effect is that the upper layer supported by the ATM VC is informed of any state change of the interface. Application cases:

- The IPoA/PPPoA/PPPoEoA PVC can be informed that the interface is down. Then, the corresponding IP interfaces are considered then as down. The result: the routes to this interface take an infinite cost, which makes it possible to re-route rapidly traffic on a backup interface.
- FRF.5: the HDLC interface can be taken down during the unavailability of the ATM PVC. This enables to signal to the connected Frame Relay equipment that the connection is not available. In this case, the connected device can send traffic on a backup interface.
- FRF.8: all DLCI supported by this ATM VC are signaled as unavailable. The signaling is achieved using LMI.

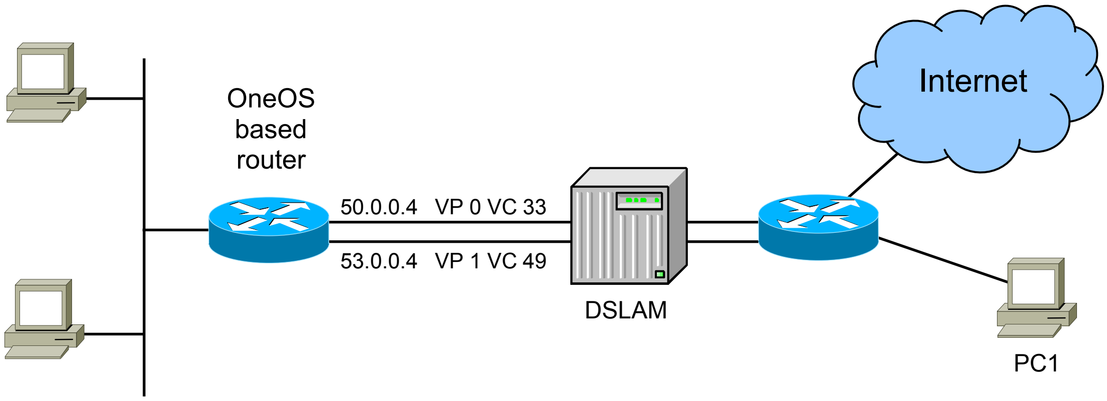
To disable intrusive ATM PVC monitoring, use the following command:

```
CLI(pvc)> no oam-pvc intrusive
```

F5 OAM intrusive management is enabled by default.

2.6.2.10 Configuration Example

In this example, the VCI 33 is used for connecting a private network to the Internet through IPoA service, while the VCI 49 is used for a PPP connection between the OneOS-based router and the remote computer named PC1.



Configuration File (CLI):

```
! ### OAM Event configuration
event filter add atm ATM_OAM ALL SHOW
configure terminal
! ### Configuration Gshdsl interface
interface atm 0
gshdsl
execute
exit
! ### Configuration ATM interface
driver ident 0
range vc min 32 max 64
range vp min 0 max 5
max vc 8
max vp 4
execute
exit
exit
! ### Configuration pvc ipoa
interface atm 0.1
pvc ipoa vpi 0 vci 33
ip address 51.0.0.4 255.255.255.0
! ### set seg-cc and ais-rdi
oam ais-rdi seg-cc
! ### validate oam configuration
oam-pvc manage
execute
exit
exit
! ### Configuration pvc pppoa
interface atm 0.2
pvc pppoa vpi 1 vci 49
ip address 53.0.0.4
ipcp static
authentication no
! ### set end-loopback and ais-rdi
oam ais-rdi end-loopback
! ### validate oam configuration and turn the pvc to intrusive
oam-pvc manage intrusive
execute
exit
exit
exit
```

2.6.2.11 Statistics

Use the following commands to display OAM statistics. To display OAM counters, OAM configuration and alarms on a VC, use the following show command:

```
CLI> show atm pvc port <atm-port> vpi <VP-identifier> vci <VC-identifier>
```

For example:

```
CLI> show atm pvc port 0 vpi 0 vci 33
OAM frequency: 10 second(s), OAM retry frequency: 1 second(s)
OAM up retry count: 3,OAM down retry count: 5
OAM CC Seg status: OK
OAM AisRdi status: enabled
OAM VC state: managed
OAM cells received: 150
F5 InEndLpbk: 0, F5 InSegLpbk: 0
F5 InEndLoop: 0, F5 InSegLoop: 0
F5 InEndAis: 0, F5 InEndRDI: 0
F5 InSegAis: 0, F5 InSegRDI: 0
F5 InEndCC: 0, F5 InSegCC: 150
OAM cells sent: 201
F5 OutEndLpbk: 0, F5 OutSegLpbk: 0
F5 OutEndLoop: 0, F5 OutSegLoop: 0
F5 OutEndAIS: 0, F5 OutEndRDI: 25
F5 OutSegAis: 0, F5 OutSegRDI: 0
F5 OutEndCC: 0, F5 OutSegCC: 176
OAM cell drops: 0
Status: UP, last Change of Status: 2507
OAM AIS down: 3, Up: 3
```

For displaying the total OAM cells sent and received on the ONE router, use the following show command:

```
CLI> show atm traffic
Total OAM cells received: 220
Total OAM cells sent: 271
```

3 ETHERNET IN THE FIRST MILE – EFM

3.1 INTRODUCING EFM

What is EFM?

Ethernet in the First Mile or EFM, also known as IEEE 802.3ah, applies Ethernet in the access network. Access networks migrate gradually from ATM based to Ethernet based: ATM switches and DSLAMs are replaced by Ethernet switches and IP DSLAMs. EFM has been defined for fiber, SHDSL and VDSL links to replace ATM also in the First Mile.

The First Mile (some call it the Last Mile) is the name traditionally given to the part of a public communication network that links the last provider-owned node (the central office or CO, the street cabinet or pole) with the customer premises equipment (CPE). The First Mile looks into it from the customer's perspective, the Last Mile from the operator’s perspective.

The Last Mile can be seen as a bottle neck in the communication network, as the bandwidth is less than in the rest of the network. EFM does not improve or replace the existing Ethernet standard; it is an extension of Ethernet technology. The EFM terminology is called 2BASE-TL when it is applied to SHDSL links and 10PASS-TS when it is applied to VDSL links.

Ethernet

Ethernet began as a broadcast local area network technology as a best effort delivery protocol. Occasional frame disruptions due to collisions or signal noise were expected and tolerated.

These days, Ethernet is omnipresent. It is easy to configure, cost-effective, highly scalable and supports a wide range of services such as data, voice and video. This makes it well suited to the demands of the First Mile, bridging the gap between the provider network and the subscriber network, making use of cable or Digital Subscriber Line (DSL).

However, quality demands in First Mile connection networks using EFM are much higher compared to LAN networks. High availability and sophisticated tools to manage and troubleshoot the EFM networks are a must for providing the high level of service customers require. Performance must be monitored, and any errors in the network must be detected and isolated very quickly.

Therefore, issues required for mass deployment of Ethernet services, such as OAM (Operation, Administration and Maintenance) and compatibility with existing technologies, have all been dealt with in the EFM standard.

More simple architecture

The use of EFM in subscriber access applications eliminates unnecessary network layers such as ATM. The elimination of network layers reduces the number of network elements in a network, and that reduces equipment costs, operational costs, and complexity.

3.2 EFM OAM – OPERATIONS, ADMINISTRATION AND MAINTENANCE

EFM OAM in accordance with IEEE 802.3ah is a mechanism that provides information, event notification, variable retrieval, and loopback controls between the CPE and the DSLAM.

The actual use of the OAM functionality is optional. A device is able to determine whether or not a remote device has the OAM functionality enabled. The OAM Discovery mechanism ascertains the configured parameters, such as maximum allowable OAM PDU size, and supported functions such as OAM remote loopback, on a given link.

For more detailed information about the OAM mechanism, refer to section 5 of IEEE Std. 802.3-2005; more specifically section 57 Operations, Administration, and Maintenance (OAM).

3.2.1 Purpose of OAM

OAM is a mechanism to:

- Monitor the functionality, link operation and health of the EFM network.
- Detect failures and improve fault isolation.
- Propagate these failures to end nodes.
- Monitor the performance of the EFM network.

The OAM mode can be set to active, passive, auto, or can be disabled. The OAM mode is further discussed below.

3.2.2 OAM PDUs or protocol data units

OAM information is conveyed in protocol frames called OAM Protocol Data Units or OAM PDUs that are sent between two ends of a single link. OAM PDUs contain the appropriate control and status information used to monitor, test and troubleshoot OAM-enabled links.

3.2.3 OAM discovery

- OAM discovery allows a local device to detect OAM on a remote device.
- Once OAM support is detected, both ends of the link exchange configuration and status information e.g. mode, PDU size, loopback support.
- If both ends are satisfied with the settings, OAM is enabled on the link.
- Loss of link and non-reception of PDUs for 5 seconds will cause the discovery process to restart.

3.2.4 OAM remote loopback

- OAM remote loopback can be used for fault localization and link performance testing. Statistics from both the local and remote DTE can be queried and compared at any time while the remote DTE is in OAM remote loopback mode.
- OAM loopback is a process that is used to verify whether a link is truly up or down. This is done by sending (and receiving) OAM Loopback PDUs between both ends of the link.
- OAM loopback can be started or stopped on the device, by using the **`oam`** command.
- The Local DTE sends loopback control PDUs, the remote DTE acknowledges by sending information PDUs with updated state information.

3.2.5 OAM Active mode

A device configured in *active* mode initiates the exchange of Information OAM PDUs. Once the Discovery process completes, active devices are permitted to send any OAM PDU while connected to a remote OAM device in *active* mode. Active devices operate in a limited respect if the remote OAM device is operating in *passive* mode. Active devices do not respond to OAM remote loopback commands and variable requests from a passive device.

A device can be set to *active* mode using the **`oam`** command, which is explained further below.

A device in *active* mode:

- Initiates the OAM Discovery process.
- Sends Information PDUs.
- May send Event Notification PDUs.
- May send Variable Request/Response PDUs.
- May send Loopback Control PDUs.

A device in *active* mode does not:

- Respond to Variable Request PDUs from other DTE in Passive mode.
- React to Loopback Control PDUs from other DTE in Passive mode.

3.2.6 OAM Passive mode

Devices configured in *passive* mode do not initiate the discovery process. Passive devices react to the initiation of the discovery process by the remote device. This eliminates the possibility of a link being established between two passive devices. Passive devices do not send Variable Request or Loopback Control OAM PDUs.

A device can be set to *passive* mode using the **`oam`** command, which is explained further below.

A device in *passive* mode:

- Waits for the remote device to initiate the discovery process.
- Sends Information PDUs.
- May send Event Notification PDUs.
- May respond to Variable Request PDUs.
- May react to Loopback Control PDUs received.

A device in *passive* mode is not permitted to send:

- Variable Request PDUs.
- Loopback Control PDUs.

3.3 EFM CONFIGURATION

First, use the following command in global configuration mode to activate the EFM interface and to enter in EFM interface configuration mode. The interface number is a logical number; it does not have a physical meaning (the interface is created if it does not exist).

```
CLI> configure terminal
CLI(configure)> interface efm <0-8>/<0-6>[.<1-255>]
CLI(conf-if)>
```

Note: As of V4.2 software release, only one EFM interface, the EFM interface 0, can be activated.

For example:

```
CLI> configure terminal
CLI(configure)> interface efm 0.1
CLI(conf-if)>
```

Second, configure the EFM interface. The available commands can be divided in 3 groups:

- Commands that are specific to G.SHDSL.bis links (refer to 3.3.1)
- Commands that are specific to VDSL2 links (refer to 3.3.2)
- Commands that apply to EFM as a standard Ethernet (FastEthernet) interface (refer to 3.3.3)

Then, use the following command to (re)start the EFM interface:

```
CLI(conf-if)> no shutdown
```

Use the following command to shutdown the EFM interface. This means that it will no longer be able to receive and send data packets. The address or addresses assigned to that interface will be unreachable. If one pings one of the interface addresses, it will not work.

```
CLI(conf-if)> shutdown
```

Finally, use the following command to exit the EFM interface configuration mode.

```
CLI(conf-if)> exit
CLI(configure)>
```

3.3.1 EFM configuration commands for G.SHDSL.bis links

Use the following command to link a DSL group to the EFM interface.

```
CLI(conf-if)> dsl-group <dsl-group-number>
```

Refer to 2.3.3.2 for more information.

Use the no form of the command to remove the DSL group from the EFM interface.

```
CLI(conf-if)> no dsl-group
```

Use the following command to set the number of line pairs (between 1 and 4; default 1) that have to be *up* for considering the whole link as up and running.

```
CLI(conf-if)> active-link-number <link-number>
```

For example: to require two line pairs to be up to consider the whole link as up.

```
CLI(conf-if)> active-link-number 2
CLI(conf-if)>
```

Use the `default` form of the command to return to the default value (1).

```
CLI(conf-if)> default active-link-number
```

Use the following command to set the minimum total speed between 0 and 22784 Kbps (sum of the speed of all lines that are up) to consider the whole link as up. The default value is 0, meaning that the link is always up.

```
CLI(conf-if)> active-user-speed <user-speed>
```

Use the `default` form of the command to return to the default value (0).

```
CLI(conf-if)> default active-user-speed
```

3.3.2 EFM configuration commands for VDSL2 links

There are no specific EFM configuration commands for VDSL2 links. As soon as the VDSL interface is up, it is automatically linked to the EFM interface. Please refer to 3.3.4 for VDSL interface configuration.

3.3.3 EFM configuration commands as an Ethernet interface

Use the following command to fill in a general description for the EFM interface. A maximum of 100 characters are allowed.

```
CLI(conf-if)> [no] description <description-of-the-interface>
```

Use the no form of the command to remove the description.

Use the following command to enable or disable OAM on the EFM interface. For more information about OAM, refer to 3.2.

```
CLI(conf-if)> oam { disable | auto | active | passive }
```

The possible values for `oam` are:

- `active`: This activates the OAM Discovery process: the OneOS-based router actively monitors the EFM link.
- `passive` (default): This sets the OAM mode to passive; this means that the OneOS-based router waits for the remote device to initiate OAM actions.
- `auto`: This means that the OneOS-based router actively monitors the EFM link AND waits for the remote device to initiate OAM actions.
- `disable`: This disables the OAM mechanism in the OneOS-based router. The EFM link will not be monitored.

All other commands are standard commands for an Ethernet interface. Refer to the ad-hoc user guide for more information about each of them.

In "OneOS – Bridging & LAN User Guide" document you will find information about the following commands:

bridge-group	in "BVI and LAN Switching" chapter
dot1Q	in "Virtual LAN and Priority Tagging" chapter

`encapsulation`	in "Virtual LAN and Priority Tagging" chapter
`ethernet`	in "Ethernet OAM" chapter
`keepalive`	in "Ethernet Physical Layer Configuration" chapter

In "OneOS – Basic IP User Guide" document you will find information about the following commands:

`arp`	in "IPv4 Configuration Commands" chapter
`bandwidth`	in "Quality of Service" chapter
`ip`	in "IPv4 Addressing and Routing" and "Internet Protocol Version 6 (IPv6)" chapters
`service-policy`	in "Quality of Service" chapter
`virtual`	in "Virtual Router Redundancy Protocol" chapter
`vrrp`	in "Virtual Router Redundancy Protocol" chapter

In "OneOS – Advanced IP User Guide" document you will find information about the following commands:

`crypto`	in "IP Security" chapter

In the present document "OneOS – WAN User Guide" you will find information about the following commands:

`pppoe`	in "PPPoE on Ethernet Uplink" chapter
`pppoe-client`	in "PPPoE on Ethernet Uplink" chapter

In order to be able to use the Virtual Router Redundancy Protocol (VRRP), this feature must be enabled on the interface. To do so, proceed as follows:

```
CLI(conf-if)> virtual interface-status
CLI(conf-if)>
```

3.3.4 VDSL

The VDSL interface is a VDSL/VDSL2 VTU-R (CPE) interface that is ITU-T G.993.1 and ITU-T G.993.2 compliant.

3.3.4.1 *VDSL Configuration*

The following command enters in VDSL interface configuration mode (`if-number` must be set to 0):

```
CLI(configure)> interface vdsl <if-number>
CLI(config-if)>
```

The VDSL interface (VTU-R) gets all its parameters from the DSLAM (VTU-O). The connection is fully automatic. As soon as the VDSL interface is activated it tries to connect.

Use the following command, in VDSL interface configuration mode, to activate the VDSL:

```
CLI(config-if)> execute
```

Then exit the VDSL interface configuration mode:

```
CLI(config-if)> exit
CLI(configure)>
```

Note: the presence of the above commands in the configuration of the OneOS-based router makes the (re)startup time one minute longer than usual startup time.

3.3.4.2 VDSL configuration example

```
interface vdsl 0
 execute
exit
interface efm 0
 ip address 10.10.10.1 255.255.255.0
exit
```

3.3.4.3 VDSL statistics

To show the VDSL interface configuration, use the following command:

```
CLI> show configuration interface vdsl 0
----------------------------------------------------------------------
interface vdsl 0 configuration:
----------------------------------------------------------------------
  minimum SNR  : 06 - def(06)
----------------------------------------------------------------------
CLI>
```

To show the VDSL interface status, use the following command:

```
CLI> show vdsl 0
--- VDSL line state -----------------------------------------------------------
State:                    showtime
Up time duration:         0d 0h 0m 13s
Up/Down counter :         13
LineType        :         VDSL2 Profile 17A
--- VDSL line status ----------------------------------------------------------
                            UPSTREAM                            DOWNSTREAM
Line Rate               :   005588 kbps (001397)              005208 kbps (001302)
Slow Payload Rate       :   004448 kbps (001112)              004448 kbps (001112)
Fast Payload Rate       :   000000 kbps (000000)              000000 kbps (000000)
Transmit Power     (dBm):     07.6
SNR Margin          (dB):    35.7
Attenuation         (dB):     24.9
--- VDSL line failure counters ------------------------------------------------
                   Near-end               Far-end
             power-on   last-read   power-on   last-read
LPR        :                        0000000000 0000000000
LOS        : 0000000013 0000000000  0000000000 0000000000
LOF        : 0000000000 0000000000  0000000000 0000000000
ESE        : 0000000000 0000000000  0000000000 0000000000
--- VDSL firmware and hardware information ------------------------------------
Firmware-VTU-R:6.5.0.3IK105012 Time Apr 23 2010, 18:52:27
RTOS 1.13.1
BME R:80 AFE<num, ver>  <0:e1>
IFE<num:Dev.Rev> <0:225.1>
CLI>
```

The VDSL statistics are viewed as follows:

```
CLI> show system statistics vdsl

====================== INI Performance Counters for Port # 1 ========================
------------------------------- ETHERNET RECEIVE -----------------------------------
 Frames...........:         3         3  Bytes............:        312        312
 Pause Frames.....:         0         0  Broadcast Frames.:          3          3
 Dropped Frames...:         0         0  Alignment Errors.:          0          0
 Oversize.........:         0         0  Undersize........:          0          0
 CRC Errors.......:         0         0  Carrier Sense Err:          0          0
------------------------------- ETHERNET TRANSMIT-----------------------------------
 Frames...........:         4         4  Bytes............:        256        256
 Pause Frames.....:         0         0  Dropped Frames...:          0          0
 Short Frames.....:         0         0  Large Frames.....:          0          0
------------------------------- ETHERNET HALF DUPLEX -------------------------------
 Late Collisions..:         0         0  Excess Collisions:          0          0
 Single Collisions:         0         0  Multi Collisions.:          0          0
 Deferred Trans...:         0         0
------------------------------------ H.D.L.C.--------------------------------------
 CRC Errors.......:         0         0  Invalid Frames...:          0          0
 Dropped Frames...:         0         0
====================================================================================

oaVdslBmeCtx[0] at 0x18499b4:
    bmeState:3 PROVISION_DONE
    bmeLoadPar:
        bmeBaseAddr: 0x42200000
        destAddr:    0x0
        pFirmware:   0x0
        firmwareLen:   995000
    bmeMemCfg:
        memType:       1
        memSize:       8
        clk:           2
        casDelay:      3
        rcDelay:       3
        rPrecharge:    3
        rActive:       7
        rCycle:        11
oaVdslPortCtx[0][0] at 0x18b14d0:
    portState         : 4 SHOWTIME
    ShowTime duration : 0d 0h 0m 26s
    adminState        : 1 UP
portId:
    portNum:                 1
    bmeNum:                  1
portStatusBrief:
    portStatus:            0x0
    FastUsPayloadRate : 0
    SlowUsPayloadRate : 1112
    bandRate          : 4000
    currentFramingMode:    0x0 ????
                      Power-UP  Last Read                            Power-UP  Last Read
NeFEC_S             0000000000 0000000000  NeFEC_F               0000000000 0000000000
NeCRC_S             0000000001 0000000000  NeCRC_F               0000000000 0000000000
NeLOS               0000000000 0000000000  NeSEF                 0000000000 0000000000
NeLOM               0000000000 0000000000  NeNCD_S               0000000000 0000000000
NeNCD_F             0000000000 0000000000  NeOCD_S               0000000000 0000000000
NeOCD_F             0000000000 0000000000  NeHEC_S               0000000000 0000000000
NeHEC_F             0000000000 0000000000  NeBER_S               0000000000 0000000000
NeBER_F             0000000000 0000000000  NeFECUnCrr_S          0000000000 0000000000
NeFECUnCrr_F        0000000000 0000000000  numUsrCell_S          0000000000 0000000000
numUsrCell_F        0000000000 0000000000  numIdlCell_S          0000000000 0000000000
numIdlCell_F        0000000000 0000000000  numTxBlk_S            0000001394 0000000401
numTxBlk_F          0000000000 0000000000  numRxBlk_S            0000001413 0000000407
numRxBlk_F          0000000000 0000000000  FeFEC_S               0000000000 0000000000
FeFEC_F             0000000000 0000000000  FeCRC_S               0000000000 0000000000
FeCRC_F             0000000000 0000000000  FeLOS                 0000000000 0000000000
FeRDI               0000000000 0000000000  FeOCD_S               0000000000 0000000000
FeOCD_F             0000000000 0000000000  FeHEC_S               0000000000 0000000000
FeHEC_F             0000000000 0000000000  FeLPR                 0000000000 0000000000
FeLOM               0000000000 0000000000  FePO                  0000000000 0000000000
FeFecEs_S           0000000000 0000000000  FeEs_S                0000000000 0000000000
FeSEs_S             0000000000 0000000000  FeLosEs_S             0000000000 0000000000
FeUnAvlEs_S         0000001365 0000000028  FeBER_S               0000000000 0000000000
UpDown              0000000013 0000000001  eocDataStr            0000000000 0000000000
FarEndCountersReq   0000000000 0000000000  FarEndCountersResp    0000000000 0000000000
ReqFarEndSnrAttn    0000000000 0000000000  RespFarEndSnrAttn     0000000000 0000000000
LoopbackReq         0000000000 0000000000  LoopbackResp          0000000000 0000000000
HeartbeatCtl        0000000000 0000000000  PtmfAddMacReq         0000000000 0000000000
PtmfRemoveMacReq    0000000000 0000000000  PtmfShowMacReq        0000000000 0000000000
PtmfReadConfigReq   0000000000 0000000000  PtmfWriteConfigReq    0000000000 0000000000
PtmfShowMacResp     0000000000 0000000000  PtmfFlushLearnAddrReq 0000000000 0000000000
PtmfConfigMacReq    0000000000 0000000000  PtmfReadConfigResp    0000000000 0000000000
RfmProtocol         0000000000 0000000000  ElmProtocol           0000000000 0000000000
ProtocolVersion     0000000000 0000000000  RmmProtocol           0000000000 0000000000
EnetStatusResp      0000000000 0000000000  VirtualMessage1     0000000000 0000000000
Other               0000000000 0000000000
Port Profile (valid only for VDSL2):
PortProfileNum              0000000001
PortProvParms:
    InterleavedChannel      0000000001
    FastChannel             0000000000
    LineType                0267452416
    DslMode                 0000000005
    RateAdaptationMode      0000000001
    InterleavedPercentage   0000000000
    UpstreamMaxRateFast     0000000000
    UpstreamMinRateFast     0000000000
    UpstreamMaxRateSlow     0000003125
    UpstreamMinRateSlow     0000000001
    UpstreamMaxInterDelay   0000000004
    DownstreamMaxRateFast   0000000000
    DownstreamMinRateFast   0000000000
    DownstreamMaxRateSlow   0000003125
    DownstreamMinRateSlow   0000000001
    DownstreamMaxInterDelay 0000000004
    NumParityFast           0000000004
    NumParitySlow           0000000004
    UsTargetNoiseMargin     0000000014
    DsTargetNoiseMargin     0000000014
    MinNoiseMargin          0000000012
    BandPlan                0000000001 BP2_998_3 (BP998_3B_8_5M)
    NumRFIBand              0000000000
    Band1Start              0000000000
    Band1Stop               0000000000
    Band2Start              0000000000
    Band2Stop               0000000000
    Band3Start              0000000000
    Band3Stop               0000000000
    Band4Start              0000000000
    Band4Stop               0000000000
    Band5Start              0000000000
    Band5Stop               0000000000
    RFIBand1                0000000000
    RFIBand2                0000000000
    RFIBand3                0000000000
    RFIBand4                0000000000
    RFIBand5                0000000000
    UsMinProtection         0000000000
    DsMinProtection         0000000000
    framingmode             0000000144
PortStartParm               0000000000
PortStopParm                0000000000
bandModifier                0000000017
PortOption                  0000000006 V43 A43 G.hs carrier set.
IFE_BRD_CFG:
    pwrMode                 0000000000 (8.5 dBm)
    txPathFilterMode        0000000000 using internal filter in Tx path
    rxPathFilterMode        0000000000 using internal filter in Rx path
Port status extended:
Downstream line rate                              5208 kbps (1302)
Upstream   line rate                              5588 kbps (1397)
Fast channel down stream payload rate             0 kbps (0)
Slow channel down stream payload rate             4448 kbps (1112)
Fast channel    upstream payload rate             0 kbps (0)
Slow channel    upstream payload rate             4448 kbps (1112)
FastDsNumParityBytes                              0
SlowDsNumParityBytes                              16
FastUsNumParityBytes                              0
SlowUsNumParityBytes                              16
Downstream delay                                  0.0 ms
Upstream   delay                                  0.0 ms
Tx total power                                    7.6 dbm
Downstream Training Margin                        36.6 dB
Downstream attainable payload rate                9886
Downstream attainable line    rate                12114
Downstream Line Protection (Slow Path)            0.0 DMT Symbols
Upstream   Training Margin                        0.0 dB
Upstream   attainable payload rate                0
Upstream   attainable line    rate                0
Upstream   Line Protection (Slow Path)            0.0 DMT Symbols
Near-end ITU Vendor Id                            0xb500494b4e530200
Far-end  ITU Vendor Id                            0xb5004946544eb184
FE Tx total power                                 10.8 dbm
VDSL Estimated Looplength                         2255
G.hs Estimated Looplength                         0
Current framing mode                              0x10 64-65 Mode (EFM)
Bandplan type: 998, 997, China, etc               0x0
No. of Upstream   Bands                           2
No. of Downstream Bands                           2
LineType                                          0x4000000 VDSL2 Profile 17A
CLI>
```

The following commands are for debugging purpose only and must be used with care.

To activate VDSL traces, use the following command:

```
CLI> trace filter add sys vdsl { all | ini | cnf | core | eoc }
                 <detail: 1-4> { show | mem | log | syslog }
```

To display the VDSL chipset event log buffer, use the following command:

```
CLI> show system statistics vdsl log
```

To save the VDSL chipset event log buffer in `/BSA/dump/BmeEventLog` file, use the following command:

```
CLI> show system statistics vdsl log file
```

3.3.4.4 ADSL/VDSL daughterboard management

On selected OneOS-based routers the same RJ11 connector can be used for either ADSL or VDSL. The ADSL and VDSL interfaces are in that case located on the same daughterboard. A relay connects automatically the line to either the ADSL interface or to the VDSL interface depending on the configuration of the router.

- When only one of the two interfaces (ADSL or VDSL) is configured, the relay is switched to the configured interface.
- When both interfaces are configured, the relay is switched first to the VDSL interface until VDSL interface is UP or a 5-minute timer elapses. When the 5-minute timer elapses, the relay is switched to the ADSL interface until ADSL interface is up or the 5-minute timer elapses again. And so on up to one interface is UP. When a line retrains (ADSL or VDSL) the relay remains on the last UP interface.

Use the following command to display the state of the relay:

```
CLI> show system statistics daughterboard avdsl
```

The state of the relay can be one of the following:

```
Idle

Switching To VDSL

Try VDSL for another <time>

VDSL is UP

Switching To ADSL

Try ADSL for another <time>

ADSL is UP
```

3.4 EFM DEBUG AND STATISTICS

To display interface statistics of a configured EFM interface, use the following command.

```
CLI(conf-if)> show interfaces efm 0
efm 0 is up, line protocol is up
  Flags: (0x8063) BROADCAST MULTICAST ARP, interface index is 5202
  Description: EFM0
  Encapsulation: Ethernet v2, MTU 1500 bytes
  Up-time 4d19h27m, status change count 1
  Hardware address is 08:00:51:03:19:91, ARP timeout 0 sec
  No auto-negotiation, half-duplex
  Line speed unknown
  Mean IP input/output rate 0/0 bits/s, 0/0 packets/s (over the last 4 seconds)
  Bridged to group 65535
  Output queuing strategy: fifo, output queue length/depth 0/126
  Reliability: 255/255
  IN:  201612 packets, 12103580 bytes, 0 queue drops
       0 broadcasts, 0 multicasts, 0 errors, 6 discards
       14 unknown protocols
  OUT: 201477 packets, 10132769 bytes, 0 queue drops
       0 broadcasts, 0 multicasts, 0 errors, 0 discards
```

To display OAM statistics of a configured EFM interface, use the following command.

```
CLI(conf-if)> show efm 0 oam

 OAM Status
 ----------

  discovery       :            fault
  loopback        :             idle

                       localInfo                  remoteInfo
                       ---------                  ----------
  version         :            1                           0
  revision        :        65535                           0
  state
    muxAction     :    forwarding                 forwarding
    parserAction  :    forwarding                 forwarding
  oui             :      00 12 EF                   00 00 00
  vendorInfo      :   00 00 00 00                00 00 00 00
  varRetrieval    :  notSupported               notSupported
  linkEvents      :   interpreted                    ignored
  loopback        :     supported               notSupported
  unidirectional  :  notSupported               notSupported
  mode            :        active                    passive
  maxPduSize      :            0                           0

 OAM Statistics
 --------------

  +------------------+--------+--------+
  |                  |   Rx   |   Tx   |
  +------------------+--------+--------+
  | information      |      0 |      0 |
  | uniqueEventNotif |      0 |      0 |
  | loopbackCtrl     |      0 |      0 |
  | varReq           |      0 |      0 |
  | varResp          |      0 |      0 |
  | orgSpecific      |      0 |      0 |
  | unsupportedCodes |      0 |      0 |
  | pduDiscard       |      0 |      0 |
  | dataDiscards     |      0 |      0 |
  +------------------+--------+--------+
```

To reset the physical layer (e.g. DSL layer) of a configured EFM interface, use the following command in global mode:

```
CLI> clear synchronize interface efm 0
```

4 SERIAL HDLC INTERFACES

Serial interfaces support HDLC-based protocols, which include Frame Relay and PPP.

The commands to enter in serial interface configuration mode and set the protocol are:

```
CLI(configure)> interface serial 0/0
CLI(config-if)> encapsulation { PPP | frame-relay }
```

4.1 V.11/V.35/V.28/V.36 SERIAL INTERFACES

The serial interface is also referred to as the Vxx interface, since the interface supports several serial protocols (V.11/V.35/V.28/V.36). Serial interfaces in the OneOS-based routers are automatically detected by the chipset by auto-detecting the connected cable. To start the Vxx Line configuration, the user must enter into the configuration mode using the command:

```
CLI> configure terminal
CLI(configure)>
```

Use the following command to enter in Vxx Line configuration mode:

```
CLI(configure)> interface serial <slot number>.<port number>
CLI(config-if)>
```

Currently, `slot number` must be set to 0 and `port number` must be set to 0.

Use the following command to enter in physical layer configuration mode:

```
CLI(config-if)> physical-layer
CLI(ph_layer)>
```

Then the user can choose:

- Either to check the Vxx Interface configuration against the detected cable type during configuration.
- Or, in case no cable is plugged-in, not to check and enter directly in physical layer configuration mode.

The command to be entered in checking mode is (default mode):

```
CLI(ph_layer)> verify
```

The command to remove the checking mode is:

```
CLI(ph_layer)> no verify
```

Some parameters can be chosen before starting the Vxx interface. All the following parameters are optional.

The following optional command specifies the Maximum Transmit Unit (MTU) used on that physical interface. The default value is 1500 bytes.

```
CLI(ph_layer)> mtu <1-1600>
```

The following optional command forces a physical modulation type in the case not to use the default modulation associated with the detected cable. **This parameter must also be provided if the device is in DCE mode and no cable is connected**.

```
CLI(ph_layer)> modulation { v28 | v35 | v11-basic | v11-rs449 | v11-rs530
                                                               | v11-rs530a }
```

The following optional command disables forced mode for physical modulation that is used (the one associated to the detected cable). It means v28 for a V.24 cable, v11-rs449 for an X.24 or V.36 cable.

```
CLI(ph_layer)> no modulation
```

By default, the physical mode DTE (terminal equipment) or DCE (circuit-terminating equipment) for the physical interface is selected during cable detection. The outgoing signals are managed as described by the chosen modulation and the cable type. The incoming checked signals that determine the state of the Vxx line are bound to modulation and cable type.

DTE mode can be chosen (optional). This allows forcing the state of outgoing signals 105 and 108 that can be forced set, forced off or managed (default) and checking the state of incoming signals 106, 107 and 109 that can be checked, not checked (ignored) or managed (default) to determine operational state of the line.

To enter in DTE mode and define the management of outgoing and incoming signals:

```
CLI(ph_layer)> dte
CLI(config-dte)> 105 { set | off | default }
CLI(config-dte)> 108 { set | off | default }
CLI(config-dte)> 106 { check | no | default }
CLI(config-dte)> 107 { check | no | default }
CLI(config-dte)> 109 { check | no | default }
CLI(config-dte)> exit
CLI(ph_layer)>
```

Use the following command to restore the default mode (DTE or DCE mode selected from the cable).

```
CLI(ph_layer)> no dte
```

DCE mode can be chosen (optional). This allows forcing the state of outgoing signals 106, 107 and 109 that can be forced set, forced off or managed (default) and checking the state of incoming signals 105 and 108 that can be checked, not checked (ignored) or managed (default) to determine operational state of the line.

To enter in DCE mode and define the management of outgoing and incoming signals:

```
CLI(ph_layer)> dce
CLI(config-dce)> 105 { check | no | default }
CLI(config-dce)> 108 { check | no | default }
CLI(config-dce)> 106 { set | off | default }
CLI(config-dce)> 107 { set | off | default }
CLI(config-dce)> 109 { set | off | default }
CLI(config-dce)> exit
CLI(ph_layer)>
```

Use the following command to restore the default mode (DTE or DCE mode selected from the cable).

```
CLI(ph_layer)> no dce
```

Use the following optional command to define the physical encoding mode:

```
CLI(ph_layer)> sync encoding { nrz | nrzi }
```

Use the following optional command to define the line rate. The default value is 2,000,000 bps.

```
CLI(ph_layer)> rate <1200-8100000>
```

Then, exit the physical layer configuration mode:

```
CLI(ph_layer)> exit
CLI(config-if)>
```

Once the physical layer is set; the user can configure the link with the upper layer.

The following commands are used:

Use the following command to set the physical interface link activity timer. The state of the interface is checked periodically (period in 1/10th of seconds). If the checked signals of the physical interface are down, the interface is considered as down.

```
CLI(config_if)> activity link timer <0-500>
```

Use the following command to set the physical interface link activity count.

```
CLI(config_if)> activity link count <0-250>
```

4.2 E1/T1 INTERFACES

To enter in E1/T1 interface configuration mode, use the following command (beginning in serial interface configuration mode):

```
CLI(config-if)> physical-layer
CLI(config-if)> g703-g704-interface
CLI(ph-layer-e1t1)>
```

Use the following command to select the alarm indication type:

```
CLI(ph-layer-e1t1)> ais-mode { itu | etsi }
```

Default: **`itu`**.

Use the following command to select the clock source (`slave` should be used when connected to a public network):

```
CLI(ph-layer-e1t1)> clock { master | slave }
```

Default: **`slave`**.

Assuming that the interface is used in structured mode (G.704/G.703), you can select the time slots that are used. The interface supports a bundle of N time slots of 64 kbps (referred to as Nx64 mode, NxPx64 is not supported):

```
CLI(ph-layer-e1t1)> ts <ts1>[-<ts2>]
```

`ts1` is the selected time slot. If you enter `ts1-ts2`, all time slots between `ts1` and `ts2` are selected. These arguments must be between 1 and 31 for E1 (1 and 24 for T1).

Example for a 1984 kbps E1 line:

```
ts 1-31
```

Example for a 1792 kbps E1 line excluding the time slots 5, 6 and 7:

```
ts 1-4
ts 8-31
```

You must specify the type of connection (E1 or T1):

```
CLI(ph-layer-e1t1)> port-type { e1 | t1 }
```

The command above makes the CLI enter in e1 or t1 configuration mode. From there, you can select some additional parameters. For E1, use:

```
CLI(ph-layer-e1)> framing { nil | ddf | mff-crc4 | mff-crc4-ext }
CLI(ph-layer-e1)> line-code { ami | hdb3 }
```

Default: **`ddf`** and **`hdb3`**.

List of Acronyms:

- `ddf`: Double Frame
- `mff-crc4`: Multi-frame with CRC4 error correction

- `mff-crc4-ext`: Extended Multi-frame with CRC4 error correction

If you have selected `port-type t1`, the parameters are:

```
CLI(ph-layer-t1)> framing { nil | f4 | f12 | esf | esf-crc6 | esf-crc6-jt
                                                                  | f72 }
CLI(ph-layer-t1)> line-code { ami | b8zs }
```

From this sub-menu, enter **exit** 3 times.

Example for 1984 kbps structured E1:

```
interface serial 0/0
 encapsulation ppp
 physical-layer
  g703-g704-interface
   ts 1-31
   port-type e1
   exit
  exit
 exit
 execute
exit
```

4.3 PPP OVER SERIAL LINE CONFIGURATION

If `encapsulation PPP` was entered in serial interface configuration mode, the next parameters must be then entered:

`profile-number`: identifier of the upper layer profile. The profile is used when defining the PPP virtual template. The value of the profile number must be between 0 and 32.

```
CLI(config-if)> profile <profile-number>
```

`execute`: The command concludes the physical interface configuration putting into place the parameters that have been set.

```
CLI(config-if)> execute
```

Example with V.11:

```
virtual-template ppp 1
 ipcp static
 ip address 10.10.10.11
 authentication no
 execute
exit

interface serial 0/0
physical layer
no verify
rate 2000000
modulation v11-rs530
dte
109 check
106 check
exit
exit
profile 1
execute
exit
```

Example with E1:

```
virtual-template ppp 1
 ipcp static
 ip address 10.10.10.11
 authentication no
 execute
exit

interface serial 0/0
 physical layer
  g703-704
   ts 1-31
   port-type e1
   exit
  exit
 exit
 profile 1
 execute
exit
```

4.3.1 Debugging PPP over Serial

To debug and decode the PPP protocol, use the following command:

```
CLI> serial-capture vxx0 [console | file] [verbose { 1 | 2 | 3 }]
```

4.4 FRAME RELAY OVER SERIAL INTERFACE

4.4.1 Features

The Frame Relay is supported on the WAN serial interface.

Fragmentation can be enabled on PVC. The fragmentation is achieved as described in the FRF.12 specification of the Frame Relay Forum. Fragmentation is recommended when real-time traffic must be transported over a multi-PVC uplink. Fragmentation prevents large fragments from non real-time PVC to block the emission of real-time packets.

The Link Fragmentation-and-Interleaving (LFI) is also supported. The LFI enables the insertion of non-fragmented frame between fragments of a large packet. It is used for example for Voice over IP/FR traffic, the frame length of which is rather short (<64 bytes). Without LFI, large IP packets prevent the emission of real-time, short packets, thus resulting unacceptable jitter and latencies. For example, a 1500-byte packet delays the emission of real-time traffic by 47 ms on a 256 kbps serial line.

The Frame Relay Traffic Shaping is based on a single token bucket. The FR traffic is shaped at the Committed Information Rate (CIR), with the following formula: `CIR = Bc/Tc`

Where `Bc` is the committed burst and `Tc` the integration period during which the scheduler computes the shaping. The depth of the token bucket is of size `Be` (Burst exceeded) that designates the number of bytes emitted beyond the CIR. Frames sent as part of the `Be` are tagged with DE=1 (Discard Eligibility). Otherwise DE=0. The `Bc`, `Be` and CIR parameters are configurable per PVC. Note that DE can also be tagged based on the DiffServ classes or routing policies.

OneOS-based routers consider the network is congested when more than 50% of the received frames are tagged with the Backward Explicit Congestion Notification (BECN). During congestion, the shaping rate is lowered, until it has reached the CIR min value.

4.4.2 Configuration

The following commands are used to create a Frame Relay (FR) PVC. A PVC is a sub-interface of the serial interface. To create a FR PVC, we must first complete the serial interface configuration as follows:

```
CLI(configure)> interface serial <slot>/<port>
CLI(config-if)> encapsulation frame-relay
```

4.4.2.1 LMI Configuration

By default, the LMI is activated in NUI mode, using the Q.933 Annex A standard. If you wish to change these parameters, the following command lines can be used:

```
CLI(config-if)> frame-relay lmi-mode { uni | nui | nni | none }
CLI(config-if)> frame-relay lmi-norm { q933 | ansi }
```

You can tune some LMI parameters:

```
CLI(config-if)> frame-relay keepalive <5-30>
```

To change the T391 DTE timer in seconds (default: 10 sec):

```
CLI(config-if)> frame-relay lmi-n391dte <1-255>
```

To change the N391DTE counter (default: 6):

```
CLI(config-if)> frame-relay lmi-n392dce <1-10>
```

To change the N392DCE counter (default: 3):

```
CLI(config-if)> frame-relay lmi-n392dte <1-10>
```

To change the N392DTE counter (default: 3):

```
CLI(config-if)> frame-relay lmi-n393dte <1-10>
```

To change the N393DTE counter (default: 4):

```
CLI(config-if)> frame-relay lmi-n393dce <1-10>
```

To change the N393DCE counter (default: 4):

For statistics about LMI, use:

```
CLI> show statistics serial <slot>/<port> [ lmi ]
```

4.4.2.2 Completing Serial Interface Configuration

When LMI is configured, complete the interface configuration by entering the next commands:

```
CLI(config-if)> execute
CLI(config-if)> exit
```

4.4.2.3 Frame-Relay QoS Configuration (Optional)

The FR QoS parameters are configured inside a class map. To create a FR class map, enter:

```
CLI(configure)> map-class frame-relay <map-class-name>
```

You can define the token bucket parameters as follows (`CIR` is mandatory):

```
CLI(config-map-class)> cir <0..200000 in kbps>
```

Optional parameters:

You can define (in bytes) the Burst Committed `Bc`. It is calculated as `CIR*Tc`, with `Tc` = 100 msec. Changing `Bc` results in changing `Tc`.

```
CLI(config-map-class)> bc <100..51200000>
```

You can define (in bytes) the Burst Exceeded `Be`. The default value is 0.

```
CLI(config-map-class)> be <0..51200000>
```

You can define (in kbps) the Minimum emission Committed Information rate for congestion periods.

```
CLI(config-map-class)> minCIR <0..200000>
```

To activate the congestion monitoring (BECN count) thereby enabling to lower the shaping rate till `MinCIR` during congestion. By default, it is not activated.

```
CLI(config-map-class)> [no] activate-shaping
```

You can define (in bytes) the size of fragments. By default fragmentation is disabled.

```
CLI(config-map-class)> fragment <16..1600>
```

The virtual QoS group is chosen for prioritization with LFI. In other words, if you want to activate LFI, you need to configure the IP QoS such that some packets are marked internally with a virtual QoS group. If a packet is marked with that group and is short enough for not being fragmented, then it can be inserted between large packet's fragments (interleaving).

```
CLI(config-map-class)> priority qos-group <0..63>
```

4.4.2.4 DLCI Configuration (Required)

Entering the following commands allows a serial sub-interface to be created for the FR PVC:

```
CLI(configure)> interface serial <slot>/<port>.<sub-interface-id>
```

`<slot>/<port>` refers to the serial port for which the FR encapsulation was previously set. The `sub-interface-id` is an arbitrary number (ranging from 1 to 255) identifying the sub-interface. It makes the CLI enter in sub-interface configuration mode. If you use an IP QoS, it must be declared here, under this sub-interface configuration level.

It is necessary to assign an IP address to the frame relay port:

```
CLI(config-if)> ip address <ip-address> [<mask>]
```

Then the user must enter a valid DLCI number for the Frame Relay circuit.

```
CLI(config-if)> frame-relay interface-dlci <dlci-value>
```

If you wish to manage Frame Relay QoS on that DLCI, apply the Frame Relay map-class defined previously:

```
CLI(config-if)> frame-relay class <map-class-name>
```

The `execute` command must be entered to validate the changes:

```
CLI(config-if)> execute
CLI(config-if)> exit
```

4.4.2.5 PVC Prioritization

To enable this feature, first enter in serial interface configuration mode and enable priority queuing:

```
CLI(configure)> interface serial <slot>.<port>
CLI(config-if)> frame-relay interface-queue priority [<1..255> <1..255>
                                                   <1..255> <1..255> ]
CLI(config-if)> execute
CLI(config-if)> exit
```

The 4 figures provided at the end of the `'frame-relay interface-queue priority'` command are optional. They represent the size of queues for the high, medium, normal and low priority queues. The default values are:

- High: 20

- Medium: 40

- Normal: 60

- Low: 80

Then, you can assign the PVC priority in the corresponding map-class as follows:

```
CLI(configure)> map-class frame-relay <map-class-name>
CLI(config-map-class)> interface-queue priority { high | medium | normal
                                                               | low }
CLI(config-if)> exit
```

Example 1: Frame Relay without QoS

```
configure terminal
interface serial 0/0
encapsulation frame-relay
execute
exit
interface serial 0/0.1
ip address 10.10.10.1 255.255.255.0
frame-relay interface-dlci 20
execute
exit
exit
```

Example 2: Frame Relay Shaping

```
configure terminal
map-class frame-relay shaper1
cir 100
bc 1250
be 500
mincir 60
adaptive-shaping
exit

interface serial 0/0
encapsulation frame-relay
execute
exit
interface serial 0/0.1
frame-relay class shaper1
ip address 10.10.10.1 255.255.255.0
frame-relay interface-dlci 20
execute
exit
exit
```

Example 3: Frame Relay Shaping with Fragmentation

```
configure terminal
map-class frame-relay shaper1
cir 100
bc 1250
be 500
mincir 60
adaptive-shaping
fragment 128
exit

interface serial 0/0
encapsulation frame-relay
execute
exit
interface serial 0/0.1
frame-relay class shaper1
ip address 10.10.10.1 255.255.255.0
frame-relay interface-dlci 20
execute
exit
exit
```

Example 4: Frame Relay Shaping with Fragmentation and LFI

```
configure terminal

! Configure a class map called 'voice'
```

```
class-map voice
match ...
exit

! Define the QoS policy to mark packets with the virtual QoS group 12
policy-map policy1
class voice
set qos-group 12
!  ...
exit
exit

map-class frame-relay shaper1
cir 100
bc 1250
be 500
mincir 60
adaptive-shaping
fragment 128
priority qos-group 12
exit

interface serial 0/0
encapsulation frame-relay
execute
exit
interface serial 0/0.1
! Apply the IP QoS policy to an interface
service-policy output policy1
frame-relay class shaper1
ip address 10.10.10.1 255.255.255.0
frame-relay interface-dlci 20
execute
exit
exit
```

4.4.3 Statistics

```
CLI> show statistics frame-relay
+---------------------------------------------------+
| IP over Frame Relay Statistics                    |
+---------------------------------------------------+
+---------------------------------------------------+
| Vxx - Port 0 / Slot 0 / Sub-Interface 1           |
| DLCI : 67                                         |
| IP address : 10.10.10.11                          |
| Status : UP                                       |
+---------------------------------------------------+
| Rx statistics                                     |
+---------------------------------------------------+
| 0 packets received                                |
| No errors                                         |
+---------------------------------------------------+
| Tx statistics                                     |
+---------------------------------------------------+
| 0 packets sent                                    |
| No errors                                         |
+---------------------------------------------------+
```

To display the traffic shaping statistics:

```
CLI> show frame-relay traffic-shaping
```

To reset the traffic shaping statistics:

```
CLI> clear frame-relay traffic-shaping
```

To show the FR configuration:

```
CLI> show frame-relay class-service
```

4.5 SERIAL LINE STATISTICS

To view information about the physical layer status, use:

```
CLI> show { e1 | vxx }
```

The following command displays the interface configuration:

```
CLI> show configuration interface serial [0/0]
```

The statistics are displayed using the following command:

```
CLI> show statistics interface serial [0/0]
```

5 PPP VIRTUAL TEMPLATE CONFIGURATION

5.1 BI-DIRECTIONAL CHAP & CHAP SERVER CONFIGURATION

When the OneOS-based router must authenticate its PPP peer with CHAP, a control is made on the username and CHAP challenge handshake. The peer username and password are defined globally and are valid for all PPP connections. When a remote device is authenticated, OneOS-based router scans the list of usernames to determine the CHAP challenge. Peers are defined as follows from the configuration terminal mode:

```
CLI(configure)> peer username <username> password <password>
                                                    [<encryption>]
```

When `encryption` is `0`, the password is entered in clear text. If `1`, it indicates the password is entered already using the encryption algorithm #1.

5.2 PPP CLIENT CONFIGURATION

First enter in PPP Virtual Template configuration mode then configure the various PPP client parameters.

5.2.1 Entering in PPP Virtual-Template Configuration Mode

The PPP Virtual template defines a profile used for the PPP over Leased Line configuration.

The virtual template profile is created by the command:

```
CLI(configure)> virtual-template ppp <profile-number>
CLI(config-virt-ppp)>
```

The parameter `profile-number` identifies the template profile (range from 0 to 32).

5.2.2 IPCP Parameters

Use the following command to configure the IPCP parameters:

```
CLI(config-virt-ppp)> ipcp { static | dynamic | no }
```

If `no` mode is entered, this feature is disabled and the IP address must then be configured.

If `static` mode is selected, the IPCP service functions in static mode. In this case the IP address must be configured and is checked by the remote equipment (the OneOS-based router and the remote device must have the same data configured –such as the IP address–).

If `dynamic` mode is selected (default value), the IP address for the interface is obtained via PPP/IPCP from the next hop router. In that case, the network mask can also be obtained via PPP/IPCP using:

```
CLI(config-virt-ppp)> [no] ipcp mask-request
```

Note: If `mask-request` is used, the `ip unnumbered <interface>` command is mandatory so that IP address and network mask apply to `<interface>`; otherwise they will apply to PPP.

If you want to retrieve the DNS server address per IPCP use the next command:

```
CLI(config-virt-ppp)> [no] ipcp dns-accept
```

Use the no form of the two above commands to not retrieve respectively the network mask and the DNS server address via PPP/IPCP.

5.2.3 IP Address

The IP address must be set if dynamic IPCP is not used. By default, dynamic IPCP is set and you do not need to set the IP address. To enter the IP address, use the following command:

```
CLI(config-virt-ppp)> ip address <ip-address> [<network-mask>]
```

Instead of having an IP address, the local PPP interface can be unnumbered, thus saving some IP addresses. The configuration is the following:

```
CLI(config-virt-ppp)> ip unnumbered <interface-type> <unit>
```

Use the no form of the above command to remove the unnumbered mode:

```
CLI(config-virt-ppp)> no ip unnumbered
```

5.2.4 Authentication Type

Use the following command to specify the authentication type.

```
CLI(config-virt-ppp)> authentication { no | pap | chap | chap/pap }
                               [ two-ways | one-way-callin | one-way-called ]
```

In case of PAP authentication, the CPE must send a name and password that is checked against a matched entry in the local database of the remote router.

In case of CHAP authentication, the CPE sends a "challenge" message to the remote device, which then encrypts the value with a shared key and returns it to the CPE.

The `authentication no` command disables this function. By default, CHAP is enabled.

The second argument defines the authentication types. They are:

- `two-ways`: authentication is done by both sides.
- `one-way-callin` (default): authentication material is sent to the remote device, but the remote device is not locally authenticated.
- `one-way-called`: authentication material is not sent to the remote device (**only username is sent**), but the remote device must be authenticated.

Note: for the current release, only bi-directional CHAP (`two-ways`) and CHAP server (`one-way-called`) are supported. All other combinations are supported with the attribute `one-way-callin`.

5.2.5 Username and Password

The following command specifies the username and password (character strings) used by the authentication protocol.

The username may content a predefined string of the form **#mac**`<i>#` that will be replaced during authentication by the corresponding MAC address of the device.

Example: `#mac1#@my.com` will be replaced by `0012EF44E276@my.com` where `00:12:EF:44:E2:76` is the MAC address #1 of the device as listed in the product information area.

The maximum permitted length of username is 64 characters; the maximum permitted length of password is 30 characters when entered in clear text and 128 characters when entered already encrypted.

```
CLI(config-virt-ppp)> username <username>
CLI(config-virt-ppp)> password <password> [<crypto-algorithm>]
```

The username must always be entered (even when authentication is `one-way-called`).

If `crypto-algorithm` is not provided, the password is entered and stored in clear text.
If `crypto-algorithm` is set to 0, the password is entered in clear text and stored encrypted.
If `crypto-algorithm` is set to 1, the password is entered and stored encrypted.

Use the following command to remove the configured password:

```
CLI(config-virt-ppp)> no password
```

5.2.6 Link Fragmentation and Interleaving (LFI)

LFI enables to reduce significantly transit delays for priority packets. To understand the under-laying properties of the IP QoS, please refer to "Transit Delay Control" section in "Quality of Service" chapter of "OneOS – Basic IP User Guide" document. When activating LFI, the priority packets are encapsulated in PPP and low priority packets are transported in MLPPP. If low priority packets are too large, they are fragmented. Priority packets are transmitted after the current frame being emitted by the interface.

To enable interleaving, use the following command:

```
CLI(config-virt-ppp)> [no] multilink interleave
```

Use the no form of the above command to disable interleaving (default value).

Then you must set some MLPPP parameters:

```
CLI(config-virt-ppp)> [no] multilink discriminator <class> <string>
```

The discriminator is used to identify to which bundle a link must be added. It must be usually configured (it depends on the remote end of the MLPPP connection); `class` is an integer ranging from 1 to 5 and `string` is a string of 20 characters maximum.

Use the no form of the above command to remove the discriminator configuration.

```
CLI(config-virt-ppp)> multilink fragment-size <bytes>

CLI(config-virt-ppp)> multilink header { long | short }

CLI(config-virt-ppp)> multilink max-rcvunit <bytes>
```

The `bytes` parameter is the Maximum Remote Receive Unit (MRRU) and is by default 1500 bytes.

5.2.7 Completing Template configuration

Please refer to 5.4 to configure other parameters that are common to PPP client and PPP server.

Then enter the following commands to complete the virtual template PPP configuration:

```
CLI(config-virt-ppp)> execute
CLI(config-virt-ppp)> exit
CLI(configure)>
```

5.2.8 Examples

The following example illustrates how to configure a PPP connection over a V11 line:

```
configure terminal
virtual-template ppp 0
ipcp static
ip address 20.0.0.2
mru local 1500 no
mru remote 1500 no
magic number not-negotiable
authentication no
keepalive 1
retry-time min 3 max 10
reconnection
```

```
execute
exit
interface serial 0/0
encapsulation ppp
profile 0
physical-layer
dte
exit
modulation v11-rs530
exit
execute
exit
```

The following example illustrates how to activate LFI over a MLPPP E1 link. We want that all PING be treated as priority (interleaved packets).

```
! Declare an ACL that matches any ICMP echo
ip access-list extended ping
 permit icmp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
exit

! Create a policy-map that puts outgoing ‘ICMP echos’ in
! the priority queue
class-map ping
 match access-group ping
exit

policy-map ping_lfi
 class ping
  priority percent 10
 exit
exit

virtual-template ppp 1
 ipcp static
 ip address 10.10.10.11
 authentication no
 multilink interleave
 multilink fragment-size 250
 multilink header long
 multilink discriminator 1 Q
 multilink max-rcvunit 1524
 execute
exit

interface serial 0/0
 physical layer
  g703-704
   ts 1-8
   port-type e1
   exit
  exit
 exit
 profile 1
 execute
 service-policy output ping_lfi
exit
```

5.3 PPP SERVER CONFIGURATION

The PPP configuration in server mode is currently restricted to application cases, where PPP sessions are tunneled in L2TP by LNS.

5.3.1 IPCP Parameters

Use the following command to configure the IPCP parameters:

```
CLI(config-virt-ppp)> ipcp { static | dynamic | no }
```

If no mode is entered, this feature is disabled and the IP address must then be configured.

If `static` mode is selected, the IPCP service, functions in static mode. In this case the IP address must be configured and is checked by the remote equipment (the OneOS-based router and the remote device must have the same data configured -such as the IP address-).

If `dynamic` mode is selected, the IP address is provided via PPP/IPCP to the remote PPP endpoint.

5.3.2 IP Address

If dynamic IPCP is used, the local PPP interface will take an IP address from the defined local pool. Two IP addresses are thus needed when a new PPP session is established. The following command must be entered, if dynamic IPCP was chosen:

```
CLI(config-virt-ppp)> ip address 0.0.0.0
```

Instead of having an IP address, the local PPP interface can be unnumbered. The configuration is the following:

```
CLI(config-virt-ppp)> ip unnumbered <interface-type> <unit>
```

The IP address must be set if dynamic IPCP is not used. To enter the IP address, use the following command:

```
CLI(config-virt-ppp)> ip address <ip-address> [<network-mask>]
```

Use the no form of the above command to remove the unnumbered mode:

```
CLI(config-virt-ppp)> no ip unnumbered
```

5.3.3 IP Address Pool

If the device is the PPP server, it can set the client IP address by taking an IP address in an address pool. The address pool is assigned as follows:

```
CLI(config-virt-ppp)> peer default ip pool <pool-name>
```

The pool must have been previously defined in global configuration mode, as follows:

```
CLI(configure)> ip local pool <pool-name> <lowest-ip-address>
                                          <highest-ip-address>
```

Use the no form of the command to remove the address pool:

```
CLI(config-virt-ppp)> no peer default ip pool
```

5.3.4 RADIUS Server

If `peer default` is enabled, the PPP authentication is realized through a RADIUS server. For authentication, a RADIUS server must be defined in global configuration mode

```
CLI(configure)> radius-server <RADIUS-server-ip> [:<RADIUS-UDP-port>]
                                                              <shared-key>
```

Example:

```
ip local pool PPPPOOL 60.3.1.1 60.3.1.50
radius-server 192.178.10.43 ip2002

virtual-template ppp 4
ipcp dynamic
ip unnumbered loopback 2
mru local 1500 no
authentication pap
keepalive 10
peer default ip pool PPPPOOL
execute
exit
```

5.3.5 Completing Template configuration

Please refer to 5.4 to configure other parameters that are common to PPP server and PPP client.

Then enter the following commands to complete the virtual template PPP configuration:

```
CLI(config-virt-ppp)> execute
CLI(config-virt-ppp)> exit
CLI(configure)>
```

5.4 COMMON PARAMETERS FOR PPP CLIENT AND PPP SERVER

5.4.1 MRU Parameters

The MRU is set using a command with the following format:

```
CLI(config-virt-ppp)> mru local <size> { yes | no } { yes | no }
CLI(config-virt-ppp)> mru remote <size> { yes | no } { yes | no }
```

This command sets the Maximum Receive Unit (MRU) local/remote length and enables selection of negotiation mode. The MRU can have different significations:

– The remote MRU designates the maximum size of packets received sent in a PPP payload. If the remote MRU is less than the IP packet length, the IP layer performs an IP fragmentation (if DF=0, otherwise the packet is dropped and an ICMP message indicating the Path MTU is sent back).

– When using MLPPP, the remote MRU sets the MLPPP fragmentation size. The size of fragments cannot be lower than 200 bytes.

If negotiation request is enabled (yes in first `yes-no` argument), the MRU length is negotiated with the remote PPP peer and is lower or equal to the size value. The negotiation accept (second `yes-no` argument) defines whether the MRU proposed by the remote peer should be accepted. If negotiation is not enabled (no in first or second `yes-no` argument), the maximum datagram length is equal to `size` value. At any rate, the length of MRU is between 40 and 9180 bytes (default value 1500 bytes).

5.4.2 Magic Number Negotiation

Magic number is used by the LCP protocol to detect looped-back lines and to check the continuity using "echo" frames.

The magic number may be negotiable: the remote device can choose the number; otherwise (if non-negotiable) the CPE selects its own number.

```
CLI(config-virt-ppp)> magic number { negotiation | not-negotiable }
```

5.4.3 Keep alive Timer

The keepalive mechanism is disabled by default. When enabled, it sends periodical "echo" frames to prevent the disconnection of the line after a long period of PPP inactivity. However, after a certain number of unsuccessful retries the line is then disconnected. Use the following command to enable the keepalive mechanism, to set the period timer and optionally to set the number of retries (20 retries by default).

```
CLI(config-virt-ppp)> [no] keepalive <timer:0-3600> [retry <1-100>]
```

Use the `no` form of the command to disable the mechanism. Note that the keepalive mechanism is also disabled when the period is set to 0.

5.4.4 Reconnection

The following command enables automatic reconnection when a PPP session is disconnected.

```
CLI(config-virt-ppp)> [no] reconnection { automatic | datagram }
```

`automatic` forces the reconnection after disconnection of the PPP session.

`datagram` forces reconnection only when a datagram needs to be forwarded on the interface.

Use the `no` form of the command to disable the mechanism. The mechanism is disabled by default.

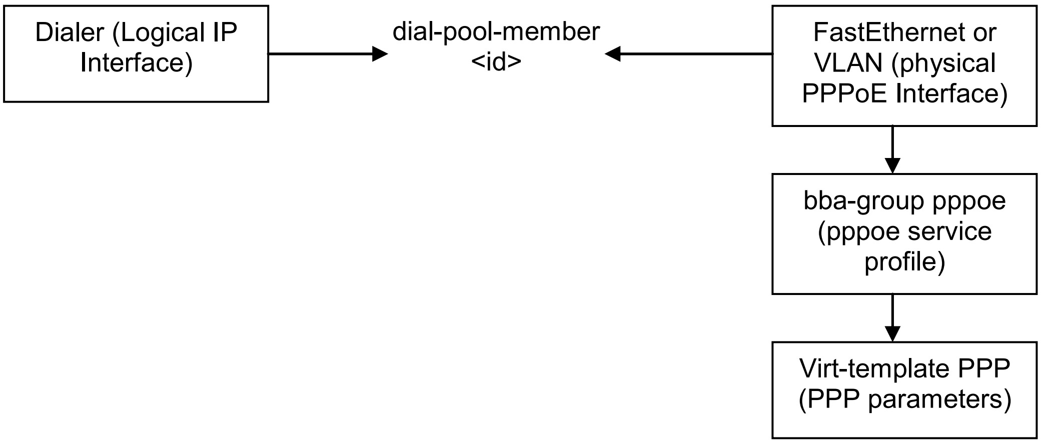
5.4.5 Retry Timer

This timer is used by the LCP protocol during the configuration and negotiation phase for connection and auto-reconnection. If a BAS is down and up again all routers will attempt to connect to the BAS simultaneously, thus flooding the BAS with connection requests. A random timer for session re-establishment can prevent several routers from reconnecting at the same time. The configuration must provide a minimum and a maximum range for retry timers. The value of the timer is then chosen randomly between the `min` and `max` values.

```
CLI(config-virt-ppp)> retry-time min <0-3600> max <0-3600>
```

6 PPPOE ON ETHERNET UPLINK

A PPPoE interface is considered in the configuration as a logical dialer interface. This dialer interface is attached to one physical or VLAN interface. The attachment between logical and physical or VLAN interface is determined by the dialer pool membership id. The physical or VLAN interface is attached to a PPPoE service profile (called `bba-group`: broadband access group), which in turn is attached to a PPP virtual template.


Dialer (Logical IP Interface)
dial-pool-member <id>
FastEthernet or VLAN (physical PPPoE Interface)
bba-group pppoe (pppoe service profile)
Virt-template PPP (PPP parameters)

The dialer interface is the IP interface onto which NAT, ACL … IP services are configured.

6.1 PPPOE CONFIGURATION

Foreword: as of V4.3R4E10 software release, multiple PPPoE sessions (up to 10) can be set on Ethernet, FastEthernet, GigabitEthernet, EFM and ATM-AAL5 interfaces as well as on corresponding sub-interfaces (VLAN interfaces). This is referred in the following as "new method". Prior to that software release only one PPPoE session could be set on only one interface listed above, this is referred as "old method".

Warning: new method and old method are exclusive each other. To switch from one method to the other method, one must erase the existing method configuration then reboot the OneOS-based router and then configure the other method.

First, select the interface or sub-interface (VLAN) where the PPPoE service shall be attached:

```
CLI> configure terminal
CLI(configure)> interface <type> <slot>/<port>[.<vlan-if>]
CLI(config-if)> no ip address
```

Then choose an arbitrary BBA group name and an arbitrary dialer pool number ID.

New method (CLI can be globally entered up to 10 times; dialer pool number IDs must be all different):

```
CLI(config-if)> pppoe enable group <bba-group-name> dialer-pool-number
                                                       <dial-pool-number>
CLI(config-if)> exit
CLI(configure)>
```

Use the no form of the command to remove the PPPoE session:

```
CLI(config-if)> no pppoe enable dialer-pool-number <id>
```

Old method (CLI can only be globally entered only 1 time):

```
CLI(config-if)> pppoe enable <bba-group-name>
CLI(config-if)> pppoe-client dial-pool-number <dial-pool-number>
CLI(config-if)> exit
CLI(configure)>
```

First, prior to configure the BBA group, a PPP virtual template is configured. This template gathers all PPP parameters for the PPPoE profile. See section 5 for more details. A typical configuration is the following:

```
CLI(configure)> virtual-template ppp <id>
CLI(config-virt-ppp)> username <ppp-username>
CLI(config-virt-ppp)> password <ppp-password>
CLI(config-virt-ppp)> ipcp dns-accept
CLI(config-virt-ppp)> execute
CLI(config-virt-ppp)> exit
CLI(configure)>
```

The BBA group is then configured. The same BBA group name must be used as the one used for the PPPoE enable command under the interface configuration mode:

```
CLI(configure)> bba-group pppoe <bba-group-name>
CLI(pppoe)>
```

Use the following command in BBA group configuration mode to apply the PPP virtual template that has been previously configured.

```
CLI(pppoe)> virtual-template <id>
```

By default, the PPPoE max MRU is 1492 bytes as per RFC 2516. Use the following command in BBA group configuration mode to allow PPPoE MTU/MRU greater than 1492 bytes according to RFC 4638. In that case, the value used is the MRU value defined in the PPP virtual template. Use the `no` form of the command to not handling MRU greater than 1492 bytes for PPPoE.

```
CLI(pppoe)> tag ppp-max-payload
```

Use the following command in BBA group configuration mode to define the PPPoE service Name. Service name is usually optional. This is an option in the PPPoE protocol that might be controlled by the BAS. If you do not know the service name, simply do not enter the command because the parameter is optional.

```
CLI(pppoe)> service-name <sv-name>
```

Terminate the BBA group configuration.

```
CLI(pppoe)> exit
CLI(configure)>
```

Then, the PPPoE dialer interface is configured. The same dial pool number must be used as the one used for the PPPoE enable command under the interface configuration mode:

```
CLI(configure)> interface dialer <id>
CLI(config-if)> encapsulation ppp
CLI(config-if)> dialer pool <dial-pool-number>
```

IP services (e.g. NAT, ACL or other IP services) are then configured as needed. For example:

```
ip nat inside overload
ip nat static-napt 192.168.1.1 21 self 21
ip access-group myacl in
```

In case dialer PPPoE is used for backup, the following command shuts down the specified dialer interface when a PADT (PPPoE Active Discovery Terminate) is received on the primary interface (refer to `padt` command in 2.5.2.3.1). The dialer specified goes back to normal state (i.e.: no shutdown state) 90 seconds after the PADT reception. This command can be entered several times in case more that one dialer PPPoE is used for backup.

```
CLI(config-if)> padt clear-pppoe-dialer <id>
```

The PPPoE session can be automatically re-initialized (disconnected then restarted) using the following command. Refer to 2.5.2.2.12 for the explanation of the parameters.

```
CLI(config-if)> reinit <down-time> <mid-time> [<up-time>]
                                                [<day-of-the-week>]
```

To restore default settings and not use automatic re-initialization of the PPPoE session, use the no form of the above command.

```
CLI(config-if)> no reinit
```

Finally, terminate the PPPoE dialer interface configuration.

```
CLI(config-if)> exit
CLI(configure)>
```

6.2 PPPOE CONFIGURATION EXAMPLE

```
hostname one100
ip name-server 127.0.0.1
policy-map qos
 class class-default
  fair-queue
 exit
exit
interface FastEthernet 1/0
 no ip address
 pppoe enable PPPoE_2
 pppoe-client dial-pool-number 15
exit
virtual-template ppp 1
 ipcp dns-accept
 username fti/9tu6kzq
 password ck934kw
 reconnection automatic
 execute
exit
bba-group pppoe PPPoE_2
 virtual-template 1
 service-name TEST_PPPOE1
exit
ip route 0.0.0.0 0.0.0.0 dialer 0
interface dialer 0
 encapsulation ppp
 dialer pool 15
 ip tcp adjust-mss 1452
 ip nat inside overload
 ip nat static-napt tcp 192.168.1.1 18080 self 80
 ip nat static-napt tcp 192.168.1.1 21 self 21
exit
```

6.3 PPPOE STATISTICS AND TRACES

To display PPP settings:

```
CLI> show virtual-template ppp <id>
```

To display PPPoE statistics:

```
CLI> show statistics dialer ppp <id>
```

To activate PPPoE traces, enter the following command and follow the guidelines:

```
CLI> [no] debug pppoe
```

To activate PPP traces, enter the following command and follow the guidelines:

```
CLI> [no] debug ppp
```

7 ISDN CONFIGURATION FOR DATA SERVICES

7.1 ISDN DATA SERVICE DESCRIPTION

The OneOS-based routers are optionally equipped with ISDN interfaces (BRI or PRI). The ISDN interface supports the LAPD signaling protocol (Q.921/Q.931) to establish the data channel supporting a PPP session. A single BRI interface has two data channels while a single PRI interface has thirty data channels. These channels are called B-channels. A PPP session can be established over a B-Channel to transport IP traffic.

B-channels can be bundled using the Multi-Link Point-to-Point Protocol (MLPPP). The MLPPP protocol permits the aggregation of several communication channels to operate as a single link; this operation is defined by RFC 1990. Using MLPPP data packets are "inverse-multiplexed" over the individual links that form the MLPPP link bundle. Latency is reduced by segmented large packets for transmission over the MLPPP link bundle.

The number of links can be increased and reduced dynamically to adapt to the required bandwidth and reduce the cost of ISDN dial-up connections. To add and remove links from an MLPPP bundle, the following is needed:

- A decision algorithm to accept/refuse link addition/removal
- (Optionally) A control protocol for negotiation of link addition/removal

The control protocol is defined in RFC 2125 and is composed of:

- The Bandwidth Allocation Protocol (BAP) which (RFC statement) "defines packets, parameters and negotiation procedures to allow two endpoints to negotiate gracefully adding and dropping links from a multilink bundle".
- The Bandwidth Allocation Control Protocol (BACP) which is the control protocol for setting up the BAP between two hosts communicating with MLPPP.

When using BAP/BACP, the calling router is the BACP client and the called router is the BACP server. The BACP client requests the addition and removal of links to the remote router. This mode is referred to as dynamic mode with a dynamic-client and dynamic-server.

If no control protocol is used, the B-Channels are added and removed by the calling router. This mode is referred to as static-boda (Bandwidth On Demand Application). Be aware that both modes are not compatible.

The following behaviors can be configured (both routers must be configured in either static or dynamic mode):

- Calling & called router (Static mode): the number of links in a bundle is defined per configuration. The device tries to add all links if they are available.
- Calling router (mode: static-boda or dynamic-client): the device can measure traffic and undertake to change the number of links to adapt to the measured traffic profile. We can define an add threshold (e.g. 80 % of the measured outgoing traffic) for adding links in a bundle. The measured traffic is a sliding average. Example: If N is the number of used links (64 kbps), if the emitted traffic exceeds (A% + (N-1)) * 64 kbps, the router tries to add a B-channel. For dropping a link, if the emitted traffic is measured as lower than (D% + (N-1)) * 64 kbps, the link is removed. In static-boda mode, the router tries to add a link as long as the measured traffic is high enough and the addition is not successful. The dynamic server can add links as long as:
    - The number of links does not exceed the number of available physical links locally and

remotely.

  - The number of used links does not exceed the configured maximum number of links for this MLPPP bundle.
  - The remote end (dynamic client) does not refuse the addition of a link to a bundle. Note: a dynamic MLPPP client can also refuse to drop links in this mode.

- Called router (mode: static-boda or dynamic-client): if the device receives a request to add/drop a link, it does not decline the request. Exception: The device can add links if available and up to a maximum number defined per configuration.

Several operating modes are possible for ISDN interface:

- Backup: when the main link fails (e.g. DSL), data are routed over the ISDN interface. Therefore, the ISDN interface is not used in normal working conditions. The ISDN interface is set up on demand, when traffic needs to be sent and no main link is operational.
- Dial-Out Access: the OneOS-based router uses the ISDN port as uplink. The access is then either permanent or dynamically set up when a datagram must be sent.
- Dial-In Access: the OneOS-based router is accessed by a remote device. The feature is useful for remote management purposes.
- Bi-directional Access: the interface can call or be called.
- PPPoA Call Back: the OneOS-based router cannot be "waked up" through the DSL uplink. The OneOS-based router receives a call setup request. Then, the ISDN call-back feature is used to reactivate the PPPoA session and the ISDN communication is not established (no B-channel is set up).

The operating modes provided above can be configured concurrently. Each of the operating modes can be configured several times with different parameters to allow several calling/called numbers. However, the services are limited by the number of B-channels. For example, one can configure Dial-Out, Dial-In and Backup accesses on a BRI interface. If the dial-out and the dial-in communication are set up, there is no B-channel left available for the backup.

For dial-on-demand connection, dialer-groups can be defined. They are a set of filtering rules that allows only certain applications to open the ISDN connection (e.g. LAN traffic, telnet) while others do not (example: SNTP, SNMP).

The configuration steps are the following:

1. ISDN Signaling Configuration (LAPD, protocol D)
2. HDLC Configuration of the B-Channels
3. ISDN Service Configuration: Dial in and/or Dial and/or backup and/or PPP call back

7.2 ISDN PHYSICAL LAYER CONFIGURATION

Note:

Use the BRI (or PRI) interface configuration mode to configure the parameters related to ISDN signaling and B-channel configuration (the "physical" layer parameters).

Then, use the isdn-dialer interface configuration mode to configure all other parameters (i.e. related to call control, IP, QoS, etc.). The use of the BRI (or PRI) interface configuration mode to configure these parameters is deprecated and must not be used.

7.2.1 ISDN Signaling Configuration

To start the ISDN configuration, the user must enter into the configuration mode using the command:

```
CLI> configure terminal
```

To create/enter the ISDN configuration, the next command is used:

```
CLI(configure)> [no] interface { bri | pri } <slot>/<port>
```

`slot` parameter is the physical slot of the ISDN interface and `port` is the physical port. The table below indicates how slots and ports are numbered on the router interfaces:

Router	Slot Number	Port Number
Single S0 ISDN daughter board (ONE60/200)	2	0
Other OneOS-based router ISDN interfaces	5	0..7-8..11(note)

Note: depending on the total number of available ISDN ports

To disable the ISDN signaling, use the `no` form of the above command.

Prior to any change in the ISDN configuration, the ISDN interface must be shut down.

```
CLI(config-if)> shutdown
```

Some parameters can be set in the ISDN signaling layer. The ISDN configuration mode is entered by entering the following command:

```
CLI(config-if)> isdn
```

The maximum number of B-channels handled by the ISDN interface is configured:

```
CLI(isdn)> max channels <number>
```

Default maximum number of handled B-channels is 2 for BRI interfaces and 30 for PRI interfaces.

The application type supported by the interface must be defined:

```
CLI(isdn)> application-interface { voip | voatm | data }
```

The default value is `voip` for voice over IP. For data services, `data` must be used. `voatm` is used on voice modules of the ONE200 for voice over ATM (BLES/VMOA).

To define the NT or TE mode in ISDN, the next commands are used:

```
CLI(isdn)> protocol-emulation { isdn-te | isdn-nt | qsig }
CLI(isdn)> layer2-emulation { nt | te }
```

`layer2-emulation` has to be defined only if it is different from `protocol-emulation`. If not defined, layer-2 emulation is the same than the protocol emulation. For BRI interfaces only, `layer1-emulation` can also be defined. If not defined, layer-1 emulation is the same than the protocol emulation.

```
CLI(isdn)> layer1-emulation { nt | te }
```

For PRI interfaces the TEI selection mode is always static and TEI is set to 0. For BRI interfaces, by default, the TEI selection mode is dynamic (recommended) and the associated command is:

```
CLI(isdn)> tei-negotiation dynamic
```

If the TEI negotiation is static, the commands are:

```
CLI(isdn)> tei-negotiation static
CLI(isdn)> static-tei <0..63>
```

Use the following command to remove the TEI also when the layer 2 is disconnected (in addition of the power-off condition):

```
CLI(isdn)> tei-remove dl-connection-release
```

Some ISDN counters and timer can be set (for experts), for example:

```
CLI(isdn)> modulo-window { 8 | 128 }
CLI(isdn)> k-window <1..128>
```

Default values are: K window = 7, modulo window = 128 (not configurable on PRI interfaces).

Then the ISDN signaling must be re-enabled using this command:

```
CLI(isdn)> exit
CLI(config-if)> no shutdown
```

7.2.2 PRI Interface Configuration

If a PRI interface is used, it must be enabled by entering the following commands, which allows specifying several parameters for the physical level.

Commands for enabling an E1 interface:

```
CLI(configure)> interface pri 5/0
CLI(config-if)> physical-interface E1
CLI(config-if)> linecode hdb3
```

Main parameters:

```
CLI(config-if)> physical-interface { E1 | T1 }
```

Specify the interface type of voice interface as E1 or T1 (default: E1):

```
CLI(config-if)> framing { none | df | mf | emf | sf | esf | auto-detect }
```

To specify the framing type:

- `none`: no framing. Only used for CES / unstructured mode
- `df`: double frame, no CRC4 (for E1 only) (default value)
- `mf`: multiframe (CRC4) (for E1 only)
- `emf`: extended multiframe (CRC4) (for E1 only)
- `sf`: super-frame (for T1 only)
- `esf`: extended super frame (for T1 only)
- `auto-detect`: used to find automatically ISDN PRI framing by trying all possible detections. This behavior can be set when the configuration of remote side is unknown. An event containing the correct framing is generated when layer-1 is activated.

Specify the physical line coding to be used `ami`, `hdb3`, `bz8s` (default: `hdb3`):

```
CLI(config-if)> linecode { ami | hdb3 | b8zs }
```

The interface must be shut down if a physical parameter has to be modified. The interface is re-activated with the command `no shutdown`.

```
CLI(config-if)> [no] shutdown
```

7.2.3 B-Channel Configuration

Usually, the next commands are not needed since the default values are generally appropriate. If some parameters need to be changed though, the user must enter in interface configuration mode:

```
CLI(configure)> interface { bri | pri } <slot>/<port>
```

The HDLC Activity Timer AT is the timer supervising the proper working of the HDLC interface. The HDLC interface is checked N times (N=HDLC count). If the timer expires N times before no HDLC flag is received, the interface is considered as faulty. To set the HDLC activity timer, use the next command:

```
CLI(config-if)> activity hdlc timer <1..500>
```

The value is provided in 10th of seconds. The default value is 36. To set the HDLC count, use the next command:

```
CLI(config-if)> activity hdlc count <1..250>
```

The default value is 1. The `inactivity b-channel` timer is a timer supervising B channel inactivity. If no traffic is used on the B channel, the B channel is disconnected after the defined timer length. The command is:

```
CLI(config-if)> inactivity b-channel <0..3600>
```

The default value is 120. The value is provided in seconds. To apply the changes:

```
CLI(config-if)> execute
CLI(config-if)> exit
```

7.3 ISDN SERVICE CONFIGURATION

7.3.1 Dial-out Calls Configuration

The Dial-Out mode permits the OneOS-based router to call a NAS or a router. If the OneOS-based router has routes with different metrics, the Dial Out can be used to trigger the interface as an ISDN backup. To configure an ISDN connection initiated by the OneOS-based device, use the next command to select an arbitrary connection number. `connection-id` is an identifier, ranging from 0 to 255, which the user can choose freely. The table in 7.2.1 indicates how slots and ports are numbered on the router interfaces.

```
CLI(configure)> interface isdn-dialer <slot>/<port>.<connection-id>
```

To enable outgoing calls, use the next command. `name` is arbitrary and optional. It designates the name of the connection such as `paris` or `central-site` for easier reading of the configuration. To disable outgoing calls and remove the ISDN dialer interface, use the no form of the command.

```
CLI(config-if)> [no] outgoing call [<name>]
```

The called E.164 number must be entered. To remove the called number, use the no form of the command.

```
CLI(out-call)> [no] address <E.164-number>
```

To configure the number of outgoing call attempts and delay between attempts, use the next command. By default, the number of retries is 0 (only one attempt) and the delay is 5 sec.

```
CLI(out-call)> call-retry <retry-number> [<seconds>]
```

Optionally, a callback number can be configured. When the calling number of an incoming call matches the callback number, the router refuses this call but rather dials the outgoing (called) address number. To remove the callback number, use the no form of the command.

```
CLI(out-call)> [no] callback-address <E.164-number>
```

By default, the delay between the incoming call and the callback call is 1 second. To change this delay (in seconds), use the next command.

```
CLI(out-call)> callback-timer <1..65535>
```

The commands can be repeated several times to allow calling different addresses.

Please refer to 7.3.4 for PPP and MLPPP parameters configuration and configuration termination.

7.3.2 Dial-in Calls Configuration (OneOS-based router being called)

The next paragraph describes the configuration steps required to allow a remote equipment to dial in the OneOS-based device. To configure the OneOS-based device to accept and establish a call initiated by a remote host, use the next command to select an arbitrary connection number. `connection-id` is an identifier, ranging from 0 to 255, which the user can choose freely. The table in 7.2.1 indicates how slots and ports are numbered on the router interfaces.

```
CLI(configure)> interface isdn-dialer <slot>/<port>.<connection-id>
```

To enable incoming calls, use the next command. `name` is arbitrary and optional. To disable incoming calls and remove the ISDN dialer interface, use the no form of the command.

```
CLI(config-if)> [no] incoming call [<name>]
```

The calling E.164 number must be entered (use "*" to disable the access control). To remove a calling address, use the no form of the command.

```
CLI(in-call)> [no] address <E.164-number>
```

The commands can be repeated several times to allow several different callers.

Please refer to 7.3.4 for PPP and MLPPP parameters configuration and configuration termination.

7.3.3 Bi-Directional Calls Configuration

To configure the OneOS-based router for an interface enabling incoming and outgoing calls, use the next command to select an arbitrary connection number. `connection-id` is an identifier, ranging from 0 to 255, which the user can choose freely. The table in 7.2.1 indicates how slots and ports are numbered on the router interfaces.

```
CLI(configure)> interface isdn-dialer <slot>/<port>.<connection-id>
```

To enable bi-directional calls, use the next command. `name` is arbitrary and optional. To disable bi-directional calls and remove the ISDN dialer interface, use the no form of the command.

```
CLI(config-if)> [no] both-way call [<name>]
```

The following E.164 numbers must be configured:

- The caller address. The caller is authenticated using this number (use "*" to disable the access control). To remove the caller number, use the no form of the command.
- The called address in case of outgoing call. To remove the called number, use the `no` form of the command.

```
CLI(bo-call)> [no] address incoming <E.164-number> [secondary]
CLI(bo-call)> [no] address outgoing <E.164-number> [secondary]
```

To configure the number of outgoing call attempts and delay between attempts, use the next command. By default, the number of retries is 0 (only one attempt) and the delay is 5 sec.

```
CLI(bo-call)> call-retry <retry-number> [<seconds>]
```

Optionally, a callback number can be configured. When the calling number of an incoming call matches the callback number, the router refuses this call but rather dials the outgoing (called) address number. To remove the callback number, use the no form of the command.

```
CLI(bo-call)> [no] callback-address <E.164-number>
```

By default, the delay between the incoming call and the callback call is 1 second. To change this delay (in seconds), use the next command.

```
CLI(bo-call)> callback-timer <1..65535>
```

The commands can be repeated several times to allow several different calls.

Please refer to 7.3.4 for PPP and MLPPP parameters configuration and configuration termination.

7.3.4 Common Calls parameters Configuration

This chapter describes the commands that are common to Dial-out calls, Dial-in calls and Bi-directional Calls. The prompt `xx-call` stands for `out-call`, `in-call` and `bo-call` prompts.

If MLPPP is used, the MLPPP bundle must be created and identified by an arbitrary identifier ranging from 0 to 32. The CLI enters in MLPPP configuration mode and some parameters can be entered.

```
CLI(xx-call)> mlppp <mlppp-id>
CLI(xx-call-mlppp)>
```

The following command defines the maximum number of links (B-channels) allowed in the bundle. Default: 1. The B-channels are taken consecutively starting with the `port-number` used in the `interface isdn-dialer` command. All B-channels have to be configured with `application-interface data`.

```
CLI(xx-call-mlppp)> max-link <1..30>
```

The following command defines the option of MLPPP for MLPPP headers (it indicates the length of sequence numbers).

```
CLI(xx-call-mlppp)> header { short | long }
```

The following command defines the number of bytes of the reassembled MLPPP payload.

```
CLI(xx-call-mlppp)> max-rcvunit <40..4000>
```

The following command defines the MLPPP discriminator (RFC 1990) that is made of a class number between 0 and 5 and an address (20 characters at most) within the class. By default class is 3 (IEEE 802.1 Globally Assigned MAC Address) and the address is the primary MAC address of the router.

```
CLI(xx-call-mlppp)> discriminator <class> <address>
```

By default OneOS apply RFC 1990 for discriminator management. That is to say a link is added in the MLPPP bundle only if the received discriminator matches the one of the existing bundle. Use the following command to disable discriminator management (disregarding RFC 1990).

```
CLI(xx-call-mlppp)> discriminator remote allow-any
```

Use the following command to re-enable discriminator management (according to RFC 1990).

```
CLI(xx-call-mlppp)> no discriminator remote allow-any
```

The following command indicates if and how the bandwidth on demand feature is activated (see 7.1):

- `static`: indicates that the number of links in the MLPPP bundle is fixed. Therefore, the device will try to set up all links when initiating the MLPPP bundle (default value).
- `static-boda`: the router measures the traffic and uses the default boda value to add and drop links. ((80% + N –1)*64 kbps, confirmation window to add: 5 sec, to remove: 1 sec)
- `dynamic-client`: the device measures bandwidth usage and add/drop links according to the measurements.
- `dynamic-server`: the device does not measure traffic.

```
CLI(xx-call-mlppp)> connection { static | static-boda | dynamic-server
                                                    | dynamic-client }
```

The following command is only available for outgoing calls (outgoing or both-way) and static-boda or dynamic-client modes. It is used to configure the bandwidth on demand parameters. Thresholds and confirmation windows define how links must be added or removed.

- A link is added if the measured inbound or outbound traffic is beyond the addition threshold. A link is removed if the measured inbound and outbound traffics are under the removal threshold.
- Thresholds: let us note a threshold T% (percentage). A sliding average of the in/out-bound traffic is calculated. To allow a link add, the measured traffic must be beyond (T% + (N-1)) x 64 kbps, where N is the number of currently open B-channels. To allow a link drop, the measured traffic must be under (T% + (N-2)) x 64 kbps.
- Confirmation Window (noted W): to avoid link flapping, a confirmation window is used. The threshold condition must be valid for W seconds to trigger the link add/drop.

Default values: add/remove threshold = 80 %, add window = 5 sec, remove window = 2 sec.

```
CLI(xx-call-mlppp)> bw-on-demand { inbound | outbound } { add | remove }
                              threshold <percent> [window <seconds>]
```

To monitor the traffic in only one direction use the next command:

```
CLI(xx-call-mlppp)> bw-on-demand {inbound | outbound} no
```

To restore default boda parameters:

```
CLI(xx-call-mlppp)> bw-on-demand default
```

Enter the following command to exit the MLPPP configuration mode

```
CLI(xx-call-mlppp)> exit
```

Then, the PPP configuration must be selected. This is done by entering the corresponding PPP virtual template identifier (a number ranging from 0 to 255):

```
CLI(xx-call)> encapsulation ppp
CLI(xx-call)> profile <virtual-template-id>
```

Then, terminate the Calls configuration:

```
CLI(xx-call)> exit
CLI(config-if)>
```

Then, apply the changes:

```
CLI(config-if)> execute
CLI(config-if)> exit
CLI(configure)>
```

7.4 ISDN STATISTICS

Note that the following commands are common to PPP and MLPPP. However, some MLPPP specific parameters are displayed when the connection uses MLPPP.

To display the ISDN interface configuration, use:

```
CLI> show configuration interface isdn
Isdn interface SLOT 5 PORT 0
    Inactivity B-channel       : 0
    HDLC supervisor timer      : 1
    HDLC supervisor count      : 3

    List of INCOMING CALL

    List of INCOMING CALL BACK

    List of OUTGOING CALL
    1   Sub interface          : 1 ,
        address out            : 2022024625
        address in             :
        Encapsulation          : PPP
        Virtual-template       : 1
        MLPPP bundle number    : 0
            Link connection mode: STATIC

            Number of link     : 2
            Bundle class number : 1
            Bundle class ref   : ZWRS
            Bundle header format: LONG
            Max rcvunit        : 4000
        Configuration status   : Completed

    List of BOTH WAY CALL
```

To display statistics for the ISDN service:

```
CLI> show statistics isdn service
------------------------------------
Common ISDN Signaling UP
------------------------------------
------------------------------------
------------------------------------
   Call ID                 : 2
   Remote E164 Address     : 2022024625
   Channel B Number        : 3
   Channel B State         : CONNECTED
   Service                 : DIRECT OUTGOING CALL
   Bri               :  5/0.1
   Service connected PPPoB OPEN
------------------------------------
------------------------------------
   Call ID                 : 3
   Remote E164 Address     : 2022024625
   Channel B Number        : 4
   Channel B State         : CONNECTED
   Service                 : MLPPP MEMBER OUTGOING CALL
   Bri               :  5/0.1
   Service connected PPPoB OPEN
```

```
---------------------------------------------------------
Isdn Backup Atm Line Ok Counter                        : 0
Isdn Backup Atm Line Refused by Network Counter        : 0
Isdn Backup Atm Line Refused by Config error Counter   : 0
-----------------------------------------------------------
-----------------------------------------------------------
Isdn Backup Serial Line Ok Counter                     : 0
Isdn Backup Atm Line Refused by Network Counter        : 0
Isdn Backup Atm Line Refused by Config error Counter   : 0
-----------------------------------------------------------
-----------------------------------------------------------
Isdn Outgoing Call Ok Counter                          : 1
Isdn Outgoing Call Callback Counter                    : 0
Isdn Callback receive                                  : 0
Isdn Outgoing Call Refused by Network Counter          : 1
Isdn Outgoing Call Refused by Config error Counter     : 0
-----------------------------------------------------------
-----------------------------------------------------------
Isdn Incoming Call Ok Counter                          : 0
Isdn Incoming Call Refused by Network Counter          : 0
Isdn Incoming Call Refused by Config error Counter     : 0
-----------------------------------------------------------
-----------------------------------------------------------
Isdn Outgoing CallBack Ok Counter                      : 0
Isdn Outgoing CallBack Refused by Network Counter      : 0
Isdn Outgoing CallBack Refused by Config error Counter : 0
```

To display PPP statistics, only for the ISDN interface:

```
CLI> show statistics isdn ppp

PPP ISDN Link Statistics
- - - - - - - - - - - - -

 Interface      : 5/0.1

NCP Statistics :
- - - - - - - -
PPP IPCP: OPEN
local address is 20.0.0.4 remote Addr is 20.0.0.1
Subnet Mask value is 32
tx packets :        2, tx bytes :       36
rx packets :        2, rx bytes :       36
packets rx > max size : 0
packets rx < min size : 0


PPP ISDN Link Statistics
- - - - - - - - - - - - -

 Interface      : 5/0.1
 Bundle number  : 0
 Link number    : 1
 Physical Layer           : Bri B Channel = 3
 E 164 Address            : 2022024625
 Service                  : Direct Outgoing Call


LCP Statistics :
- - - - - - - -
PPP LCP: OPEN
Local MRU: 1500, Remote MRU: 1500
tx packets :        2, tx bytes :       51
rx packets :        2, rx bytes :       51
packets rx > max size : 0
packets rx < min size : 0
tx packets Config Req     :        1, rx packets Config Req     :        1
tx packets Config Acq     :        1, rx packets Config Acq     :        1
tx packets Config Nak     :        0, rx packets Config Nak     :        0
tx packets Config Rej     :        0, rx packets Config Rej     :        0
tx packets Termination Req:        0, rx packets Termination Req:        0
tx packets Termination Acq:        0, rx packets Termination Acq:        0
tx packets Code Rej       :        0, rx packets Code Rej       :        0
tx packets Protocol Rej   :        0, rx packets Protocol Rej   :        0
tx packets Echo Req       :        0, rx packets Echo Req       :        0
tx packets Echo Rep       :        0, rx packets Echo Rep       :        0
tx packets Discard Req    :        0, rx packets Discard Req    :        0
tx packets Identification :        0, rx packets Identification :        0
tx packets Time remaining :        0, rx packets Time remaining :        0


PPP ISDN Link Statistics
- - - - - - - - - - - - -

 Interface      : 5/0.1
 Bundle number  : 0
 Link number    : 2
 Physical Layer           : Bri B Channel = 4
 E 164 Address            : 2022024625
 Service                  : Direct Outgoing Call


LCP Statistics :
- - - - - - - -
PPP LCP: OPEN
Local MRU: 1500, Remote MRU: 1500
tx packets :        2, tx bytes :       51
rx packets :        2, rx bytes :       51
packets rx > max size : 0
packets rx < min size : 0
tx packets Config Req     :        1, rx packets Config Req     :        1
tx packets Config Acq     :        1, rx packets Config Acq     :        1
tx packets Config Nak     :        0, rx packets Config Nak     :        0
tx packets Config Rej     :        0, rx packets Config Rej     :        0
tx packets Termination Req:        0, rx packets Termination Req:        0
tx packets Termination Acq:        0, rx packets Termination Acq:        0
tx packets Code Rej       :        0, rx packets Code Rej       :        0
tx packets Protocol Rej   :        0, rx packets Protocol Rej   :        0
tx packets Echo Req       :        0, rx packets Echo Req       :        0
tx packets Echo Rep       :        0, rx packets Echo Rep       :        0
tx packets Discard Req    :        0, rx packets Discard Req    :        0
tx packets Identification :        0, rx packets Identification :        0
tx packets Time remaining :        0, rx packets Time remaining :        0
```

To display aggregate MLPPP statistics:

```
CLI> show statistics isdn service mlppp
NumBundle         = 0
LinkNumber        = 2
Slot, Port, Vcd   = 5, 0, 1

LinkNumber | LinkType  | CallId | BChannel Num | State      |
          0| PRINCIPAL |       2|            3| ACTIVE     |
          1| SECONDARY |       3|            4| ACTIVE     |
```

To display MLPPP bundle statistics (with physical ports):

```
CLI> show isdn mlppp
isdn-dialer 5/0.1 , bundle number is 0
 username is oneaccess
 Endpoint discriminator is OA-BU
 Rx-Reorder
  0 fragments in reorder list
  0 lost fragments
  4 reordered
  0 discarded included 0 lost received
  5 expected sequence nb
 Bundle up for 00:00:32
 Max Link number : 8
 Active Links : 8
  BRI 5/0:0 for 00:00:32
  BRI 5/0:1 for 00:00:26
  BRI 5/1:0 for 00:00:26
  BRI 5/1:1 for 00:00:26
  BRI 5/2:0 for 00:00:26
  BRI 5/2:1 for 00:00:26
  BRI 5/3:0 for 00:00:26
  BRI 5/3:1 for 00:00:26
```

To display the list of active (connected) calls using ISDN ports:

```
CLI> show isdn active

--------------------------------------------------------------------------------
                               ISDN ACTIVE CALLS
--------------------------------------------------------------------------------
App.  Call         Calling           Called   Call   Port BChan Call-ref call-id
      Type          Number           Number Duration
--------------------------------------------------------------------------------
 Data Out                        2123456789 00:00:33  5/0   1        3        9
 Data Out                        2123456789 00:00:27  5/0   2        4       13
 Data Out                        2123456789 00:00:27  5/1   1        3       10
 Data Out                        2123456789 00:00:27  5/1   2        4       14
 Data Out                        2123456789 00:00:27  5/2   1        3       11
 Data Out                        2123456789 00:00:27  5/2   2        4       15
 Data Out                        2123456789 00:00:27  5/3   1        3       12
 Data Out                        2123456789 00:00:27  5/3   2        4       16
--------------------------------------------------------------------------------
```

To display the protocol status on ISDN ports:

```
CLI> show isdn status [<slot>/<port> | all]
```

```
isdn line                       5/0
       protocol descriptor     BRI_TE
       config state            down
       layer 1 status          activated
       layer 2 status          activated
                               tei= 65 , ces=  1 , state= established
       layer 3 status
           active call         2
                -> App= Data , BC=, type=Out , BChan= 1 , call-ref= 1 , call-id= 1
                -> App= Data , BC=, type=Out , BChan= 2 , call-ref= 2 , call-id= 5
```

To display ISDN sub-interface IP statistics:

```
CLI> show interfaces isdn [<slot>/<port>.<sub-if>]
Isdn 5/0.1 is up, line protocol is dormant
  Flags: (0x80f1) POINT-TO-POINT MULTICAST, interface index is 15601
  Encapsulation: Point-to-Point Protocol, MTU 1500 bytes
  Dormant-time 00:00:06, status change count 2
  Internet address is 20.0.0.4/32
  Line speed 64 kbps
  Mean IP input/output rate 0/0 bits/s, 0/0 packets/s (over the last 4 seconds)
  Output queuing strategy: cbq
  Reliability: 255/255
  IN:  15 packets, 1800 bytes, 0 queue drops
       0 broadcasts, 0 multicasts, 0 errors, 0 discards
       0 unknown protocols
  OUT: 15 packets, 1800 bytes, 0 queue drops
       0 broadcasts, 0 multicasts, 0 errors, 0 discards
```

Note: the next deprecated command does nothing and is replaced by above and following commands.

```
CLI>show system statistics isdn-connections
```

To display detailed statistics for the BRI B channels, use:

```
CLI> show system statistics bri
ISDN BRI Show
Bundle 5/0.1 : 8 active channels TX/RX bytes 1830/1802 (polled 4626)
 POOL ID = 0x0
     Inactivity Configured 120 Current 110
 ISDN [BRI1] Line:5/0 Channel B1 Status ACTIVE UP (5/0.1)
   TX Pckts submitted    NORM        14 PRIO         0 TOTAL          14
      Pckts pending cur. NORM         0 PRIO         0 TOTAL           0
      Pckts pending max. NORM         3 PRIO         0 Qerror          0
      Transmitter busy            4   Polling KO          0
      HDLC Pckts   Start         14   Sent               14
        INTR XPR(frame)               14   XPR(pool)          20
        INTR XDU(underrun)             0   XPR(reset)          1
        HSCX send ko state             0 ko xfifo            0
   RX Pckts  OK                  14 (RX Bytes       1382)
        RX Packets discarded           0 (no buffers)
        INTR RME total                14 RPF total         20
        INTR RME nomem                 0 RPF nomem          0
        INTR RME Errors RDO            0 RME Errors CRC          0
        INTR RME Errors RAB            0 RME Errors SIZE         0
        INTR RFO Overflow              0
     Inactivity Configured 120 Current 112
 ISDN [BRI1] Line:5/0 Channel B2 Status ACTIVE UP (5/0.1)
   TX Pckts submitted    NORM         2 PRIO         0 TOTAL           2
      Pckts pending cur. NORM         0 PRIO         0 TOTAL           0
      Pckts pending max. NORM         1 PRIO         0 Qerror          0
      Transmitter busy            0   Polling KO          0
      HDLC Pckts   Start          2   Sent                2
        INTR XPR(frame)                2   XPR(pool)           0
        INTR XDU(underrun)             0   XPR(reset)          1
        HSCX send ko state             0 ko xfifo            0
   RX Pckts  OK                   2 (RX Bytes         52)
        RX Packets discarded           0 (no buffers)
        INTR RME total                 2 RPF total          0
        INTR RME nomem                 0 RPF nomem          0
        INTR RME Errors RDO            0 RME Errors CRC          0
        INTR RME Errors RAB            0 RME Errors SIZE         0
        INTR RFO Overflow              0
... (other ISDN Lines) ...
*************** Rx-reorder Configuration **************
 MLPPPoB Header mode               : LONG
 RMax                              : [50]
 Max Polling                       : [15]
 Late Packet Politic               : DROP
*******************************************************
*************** Rx-reorder Statistics *****************
---------------------- Treated ------------------------
Nb Packets forwarded in sequence         : [        10]
Nb Packets Dropped (because Late)       : [         0]
Nb LCP Packets Bundle (Dropped)         : [         0]
Nb Packets Dropped at Ring init         : [         0]
Nb Packets Dropped (internal Error)     : [         0]
Nb Late Packets Dropped (internal Error): [         0]
Nb Packets declared loss                : [         0]
Nb Packets Mean by job                  : [         1]
Nb job done (pppEndSwitchJob2 call)     : [        10]
Nb netJobAdd done                       : [        10]
Nb netJobAdd failed                     : [         0]
------------------------ Rx ----------------------------
Nb Packets Rx From Drivers         : [        28]
   - SIG PPP Packets               +>: [        16]
   - Sequenced packets             +>: [        12]
     - IPCP Packets                | +>: [         2]
     - Data packets MLPPPoB        | +>: [        10]
       - Consider as LATE Packets  |   +>: [         0]
       - Sequenced Packets         |   +>: [        10]
       - Ring Full ( Dropped )     |   +>: [         0]
   - Dropped packets               +>: [         0]
     - Stack is NULL                 +>: [         0]
     - Mblk is NULL                  +>: [         0]
     - Channel is Down               +>: [         0]
-------------------------------------------------------
**************** Rx-reorder Status *******************
Nb Packet in the Ring              : [0]
Next expected MLPPP sequence nb    : [12]
Sequencer state : STOPPED
*******************************************************
```

To display aggregated statistics for the BRI B channels, use:

```
CLI> show system statistics bri short
|   TX pending      |   TX     |   TX     |   RX     |
|  PRIO   |  NORM   |            |          |          |
|Curr|Max |Curr|Max |Tx Qerror | Tx Done  | Rx Done  |
|   0|   0|   0|   3|         0|        14|        14|
|   0|   0|   0|   1|         0|         2|         2|
|   0|   0|   0|   1|         0|         2|         2|
|   0|   0|   0|   1|         0|         2|         2|
|   0|   0|   0|   1|         0|         2|         2|
|   0|   0|   0|   1|         0|         2|         2|
|   0|   0|   0|   1|         0|         2|         2|
|   0|   0|   0|   1|         0|         2|         2|
Bundle 5/0.1 : 8 active channels TX/RX bytes 1830/1802 (polled 12472)
       [TX] 0 bps  [RX] 0 bps (last 31 seconds)
```

To reset statistics for the BRI B channels, use:

```
CLI> show system statistics bri reset
```

To display information about physical BRI B channel structures, use:

```
CLI> show system statistics bri phb
BRI devices and channels
oaBriDevCtx 0x015269c4 iDevice = 0
 pDevBriCtx->DevInitDone = 0
 pDevBriCtx->DevId       = 0
 pDevBriCtx->NaiId       = 255
 pDevBriCtx->IsacId      = 0
 pDevBriCtx->DevName     = BRI0
 pDevBriCtx->BriNetPool  = 0x00000000
 ======= pPhbCtx 0x01526a08 iChannel = 0
  pPhbCtx->BCtxApiState   0
  pPhbCtx->BCtxB1B2       0
  pPhbCtx->BCtxMagic      0xdead7654
  pPhbCtx->BctxHandle     0x01526a08
  pPhbCtx->BctxFastHandle 0x00000000
  pPhbCtx->BCtxConfig     0x0272f698
  pPhbCtx->BCtxDevCtx     0x015269c4
  pPhbCtx->BType          1
  pPhbCtx->BCtxToBundle   0x00000000
  &pPhbCtx->BCtxStats     0x01526a7c
  pPhbCtx->BCtxConfig.slot/port/vcd     0/0/0
  pPhbCtx->BCtxConfig.inactivity        0
  pPhbCtx->BCtxConfig.ChannelIsSingle   0
  pPhbCtx->BCtxConfig.ChannelRank       0
  pPhbCtx->BCtxConfig.ChannelInterleave 0
  pPhbCtx->BCtxConfig.ChannelFragmentSz 0
 ======= pPhbCtx 0x01526b08 iChannel = 1
  pPhbCtx->BCtxApiState   0
  pPhbCtx->BCtxB1B2       1
  pPhbCtx->BCtxMagic      0xdead7654
  pPhbCtx->BctxHandle     0x01526b08
  pPhbCtx->BctxFastHandle 0x00000000
  pPhbCtx->BCtxConfig     0x0272f698
  pPhbCtx->BCtxDevCtx     0x015269c4
  pPhbCtx->BType          1
  pPhbCtx->BCtxToBundle   0x00000000
  &pPhbCtx->BCtxStats     0x01526b7c
  pPhbCtx->BCtxConfig.slot/port/vcd     0/0/0
  pPhbCtx->BCtxConfig.inactivity        0
  pPhbCtx->BCtxConfig.ChannelIsSingle   0
  pPhbCtx->BCtxConfig.ChannelRank       0
  pPhbCtx->BCtxConfig.ChannelInterleave 0
  pPhbCtx->BCtxConfig.ChannelFragmentSz 0
... (other BRI devices and channels) ...
PRI devices and channels
oaPriDevCtx 0x015e452c iDevice = 0
 pDevPriCtx->DevInitDone = 0
 pDevPriCtx->DevId       = 0
 pDevPriCtx->DevName     = PRI0
 pDevPriCtx->BriNetPool  = 0x00000000
 ======= pPhbCtx 0x015e4570 iChannel = 0
  pPhbCtx->BCtxApiState   0
  pPhbCtx->BCtxB1B2       0
  pPhbCtx->BCtxMagic      0xdead7654
  pPhbCtx->BctxHandle     0x015e4570
  pPhbCtx->BctxFastHandle 0x00000000
  pPhbCtx->BCtxConfig     0x02c100e0
  pPhbCtx->BCtxDevCtx     0x015e452c
  pPhbCtx->BType          3
  pPhbCtx->BCtxToBundle   0x00000000
  &pPhbCtx->BCtxStats     0x015e45e4
  pPhbCtx->BCtxConfig.slot/port/vcd     0/0/0
  pPhbCtx->BCtxConfig.inactivity        0
  pPhbCtx->BCtxConfig.ChannelIsSingle   0
  pPhbCtx->BCtxConfig.ChannelRank       0
  pPhbCtx->BCtxConfig.ChannelInterleave 0
  pPhbCtx->BCtxConfig.ChannelFragmentSz 0
... (other PRI devices and channels) ...
oaBriPhbNetBricksCtxShow:
oaBriPhbNetBricksInit : network adapter objects
First NAI:
NA 0 (0x02c3ba48)
nai                   = 0x0
hscx_id               = 0x0
audio_law             = 0x41
hscx_type             = 0x2
p_mem_blk             = 0x2c3ba40
ent_up_default        = 0x6e
size                  = 0x22c
  Channel 0 (0x02c3ba54)
  chan_nai            = 0x0
  chani               = 0x0
  a_peitab            = 0x2c3ba54
  ts_tx               = 0x0
  ts_rx               = 0x0
  init_mode           = 0x4f
  ent_up              = 0x6e
  layer_3_ent_id      = 0x6e
  wait_commit_flag    = 0x0
  error_counter       = 0x0
  congestion          = 0x0
  ts_size             = 0x8
...
```

To display statistics for MLPPP over BRI B channel sequencer, use:

```
CLI> show system statistics bri sequencer
*************** Rx-reorder Configuration **************
 MLPPPoB Header mode               : LONG
 RMax                              : [50]
 Max Polling                       : [15]
 Late Packet Politic               : DROP
*******************************************************
*************** Rx-reorder Statistics *****************
---------------------- Treated ------------------------
Nb Packets forwarded in sequence         : [        10]
Nb Packets Dropped (because Late)       : [         0]
Nb LCP Packets Bundle (Dropped)         : [         0]
Nb Packets Dropped at Ring init         : [         0]
Nb Packets Dropped (internal Error)     : [         0]
Nb Late Packets Dropped (internal Error): [         0]
Nb Packets declared loss                : [         0]
Nb Packets Mean by job                  : [         1]
Nb job done (pppEndSwitchJob2 call)     : [        10]
Nb netJobAdd done                       : [        10]
Nb netJobAdd failed                     : [         0]
------------------------ Rx ----------------------------
Nb Packets Rx From Drivers         : [         28]
   - SIG PPP Packets               +>: [         16]
   - Sequenced packets             +>: [         12]
```

```
     - IPCP Packets                | +>: [         2]
     - Data packets MLPPPoB        | +>: [        10]
       - Consider as LATE Packets  |   +>: [         0]
       - Sequenced Packets         |   +>: [        10]
       - Ring Full ( Dropped )     |   +>: [         0]
   - Dropped packets               +>: [         0]
     - Stack is NULL                 +>: [         0]
     - Mblk is NULL                  +>: [         0]
     - Channel is Down               +>: [         0]
-------------------------------------------------------
***************** Rx-reorder Status *******************
Nb Packet in the Ring              : [0]
Next expected MLPPP sequence nb    : [12]
Sequencer state : STOPPED
*******************************************************
```

To reset statistics for MLPPP over BRI B channel sequencer, use:

```
CLI> show system statistics bri sequencer reset
```

To display detailed statistics for the PRI B channels, use:

```
CLI> show system statistics pri
Bundle 5/0.1 : 30 active channels TX/RX bytes 9562/10010 (polled 21901)
 POOL ID = 0x0
     Inactivity Configured 120 Current 104
 ISDN [PRI0] Line:5/0 Channel B1 Status ACTIVE UP (5/0.1)
   TX Pckts submitted    NORM         28 PRIO          0 TOTAL          28
      Pckts pending cur. NORM          0 PRIO          0 TOTAL           0
      Pckts pending max. NORM          5 PRIO          0 Qerror          0
      Transmitter busy           0   Polling KO          0
      HDLC Pckts   Start         28   Sent               28
   RX Pckts  OK                  24 (RX Bytes       1624)
        RX Packets discarded            0 (no buffers)
        INTR RME Errors RDO             0 RME Errors CRC           0
        INTR RME Errors RAB             0 RME Errors SIZE          0
        INTR RFO Overflow               0
     Inactivity Configured 120 Current 106
 ISDN [PRI0] Line:5/0 Channel B2 Status ACTIVE UP (5/0.1)
   TX Pckts submitted    NORM         11 PRIO          0 TOTAL          11
      Pckts pending cur. NORM          0 PRIO          0 TOTAL           0
      Pckts pending max. NORM          1 PRIO          0 Qerror          0
      Transmitter busy           0   Polling KO          0
      HDLC Pckts   Start         11   Sent               11
   RX Pckts  OK                  13 (RX Bytes        493)
        RX Packets discarded            0 (no buffers)
        INTR RME Errors RDO             0 RME Errors CRC           0
        INTR RME Errors RAB             0 RME Errors SIZE          0
        INTR RFO Overflow               0
     Inactivity Configured 120 Current 107
 ISDN [PRI0] Line:5/0 Channel B3 Status ACTIVE UP (5/0.1)
   TX Pckts submitted    NORM         11 PRIO          0 TOTAL          11
      Pckts pending cur. NORM          0 PRIO          0 TOTAL           0
      Pckts pending max. NORM          1 PRIO          0 Qerror          0
      Transmitter busy           0   Polling KO          0
      HDLC Pckts   Start         11   Sent               11
   RX Pckts  OK                  13 (RX Bytes        493)
        RX Packets discarded            0 (no buffers)
        INTR RME Errors RDO             0 RME Errors CRC           0
        INTR RME Errors RAB             0 RME Errors SIZE          0
        INTR RFO Overflow               0
     Inactivity Configured 120 Current 107
. . .
ISDN [PRI0] Line:5/0 Channel B15 Status ACTIVE UP (5/0.1)
   TX Pckts submitted    NORM         11 PRIO          0 TOTAL          11
      Pckts pending cur. NORM          0 PRIO          0 TOTAL           0
      Pckts pending max. NORM          1 PRIO          0 Qerror          0
      Transmitter busy           0   Polling KO          0
      HDLC Pckts   Start         11   Sent               11
   RX Pckts  OK                  13 (RX Bytes        493)
        RX Packets discarded            0 (no buffers)
        INTR RME Errors RDO             0 RME Errors CRC           0
        INTR RME Errors RAB             0 RME Errors SIZE          0
        INTR RFO Overflow               0
     Inactivity Configured 120 Current 107
 ISDN [PRI0] Line:5/0 Channel B17 Status ACTIVE UP (5/0.1)
   TX Pckts submitted    NORM         11 PRIO          0 TOTAL          11
      Pckts pending cur. NORM          0 PRIO          0 TOTAL           0
      Pckts pending max. NORM          1 PRIO          0 Qerror          0
      Transmitter busy           0   Polling KO          0
      HDLC Pckts   Start         11   Sent               11
   RX Pckts  OK                  13 (RX Bytes        493)
        RX Packets discarded            0 (no buffers)
        INTR RME Errors RDO             0 RME Errors CRC           0
        INTR RME Errors RAB             0 RME Errors SIZE          0
        INTR RFO Overflow               0
     Inactivity Configured 120 Current 108
. . .
ISDN [PRI0] Line:5/0 Channel B29 Status ACTIVE UP (5/0.1)
   TX Pckts submitted    NORM          9 PRIO          0 TOTAL           9
      Pckts pending cur. NORM          0 PRIO          0 TOTAL           0
      Pckts pending max. NORM          1 PRIO          0 Qerror          0
      Transmitter busy            0   Polling KO          0
      HDLC Pckts   Start          9   Sent                9
   RX Pckts  OK                  11 (RX Bytes        443)
        RX Packets discarded            0 (no buffers)
        INTR RME Errors RDO             0 RME Errors CRC           0
        INTR RME Errors RAB             0 RME Errors SIZE          0
        INTR RFO Overflow               0
     Inactivity Configured 120 Current 99
 ISDN [PRI0] Line:5/0 Channel B30 Status ACTIVE UP (5/0.1)
   TX Pckts submitted    NORM          9 PRIO          0 TOTAL           9
      Pckts pending cur. NORM          0 PRIO          0 TOTAL           0
      Pckts pending max. NORM          1 PRIO          0 Qerror          0
      Transmitter busy            0   Polling KO          0
      HDLC Pckts   Start          9   Sent                9
   RX Pckts  OK                  11 (RX Bytes        443)
        RX Packets discarded            0 (no buffers)
        INTR RME Errors RDO             0 RME Errors CRC           0
        INTR RME Errors RAB             0 RME Errors SIZE          0
        INTR RFO Overflow               0
     Inactivity Configured 120 Current 99
 ISDN [PRI0] Line:5/0 Channel B31 Status ACTIVE UP (5/0.1)
   TX Pckts submitted    NORM         10 PRIO          0 TOTAL          10
      Pckts pending cur. NORM          0 PRIO          0 TOTAL           0
      Pckts pending max. NORM          1 PRIO          0 Qerror          0
      Transmitter busy            0   Polling KO          0
      HDLC Pckts   Start         10   Sent               10
   RX Pckts  OK                  12 (RX Bytes        468)
        RX Packets discarded            0 (no buffers)
        INTR RME Errors RDO             0 RME Errors CRC           0
        INTR RME Errors RAB             0 RME Errors SIZE          0
        INTR RFO Overflow               0
*************** Rx-reorder Configuration **************
 MLPPPoB Header mode               : LONG
 RMax                              : [50]
 Max Polling                       : [15]
 Late Packet Politic               : DROP
*******************************************************
*************** Rx-reorder Statistics *****************
---------------------- Treated ------------------------
Nb Packets forwarded in sequence         : [       103]
Nb Packets Dropped (because Late)       : [         0]
Nb LCP Packets Bundle (Dropped)         : [         0]
Nb Packets Dropped at Ring init         : [         0]
Nb Packets Dropped (internal Error)     : [         0]
Nb Late Packets Dropped (internal Error): [         0]
Nb Packets declared loss                : [         0]
Nb Packets Mean by job                  : [         1]
Nb job done                             : [       103]
------------------------ Rx ----------------------------
Nb Packets Rx From Drivers         : [       249]
   - SIG PPP Packets               +>: [       144]
   - Sequenced packets             +>: [       105]
     - IPCP Packets                | +>: [         2]
     - Data packets MLPPPoB        | +>: [       103]
       - Consider as LATE Packets  |   +>: [         0]
       - Sequenced Packets         |   +>: [       103]
       - Ring Full ( Dropped )     |   +>: [         0]
   - Dropped packets               +>: [         0]
     - Stack is NULL                 +>: [         0]
     - Mblk is NULL                  +>: [         0]
     - Channel is Down               +>: [         0]
-------------------------------------------------------
***************** Rx-reorder Status *******************
Nb Packet in the Ring              : [0]
Next expected MLPPP sequence nb    : [106]
*******************************************************
```

To display aggregated statistics for the PRI B channels, use:

```
CLI> show system statistics pri short
|   TX pending      |   TX     |   TX     |   RX     |
|  PRIO   |  NORM   |          |          |          |
|Curr|Max |Curr|Max |Tx Qerror | Tx Done  | Rx Done  |
|   0|   0|   0|   5|         0|        29|        25|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        11|        13|
|   0|   0|   0|   1|         0|        11|        13|
|   0|   0|   0|   1|         0|        10|        12|
|   0|   0|   0|   1|         0|        10|        12|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        12|        14|
|   0|   0|   0|   1|         0|        11|        13|
|   0|   0|   0|   1|         0|        10|        12|
|   0|   0|   0|   1|         0|        10|        12|
|   0|   0|   0|   1|         0|        11|        13|
Bundle 5/0.1 : 30 active channels TX/RX bytes 12730/13134 (polled 32988)
       [TX] 568 bps  [RX] 560 bps (last 44 seconds)
```

To reset statistics for the PRI B channels, use:

```
CLI> show system statistics pri reset
```

To display information about physical PRI B channel structures, use:

```
CLI> show system statistics pri phb
PRI devices and channels
oaPriDevCtx 0x01707f40 iDevice = 0
 pDevPriCtx->DevInitDone = 1
 pDevPriCtx->DevId       = 0
 pDevPriCtx->DevName     = PRI0
 pDevPriCtx->BriNetPool  = 0x03052da0
 ======= pPhbCtx 0x01707f84 iChannel = 0
  pPhbCtx->BCtxApiState   0
  pPhbCtx->BCtxB1B2       0
  pPhbCtx->BCtxMagic      0xdead7654
  pPhbCtx->BctxHandle     0x01707f84
  pPhbCtx->BctxFastHandle 0x00000000
  pPhbCtx->BCtxConfig     0x0307c828
  pPhbCtx->BCtxDevCtx     0x01707f40
  pPhbCtx->BType          3
  pPhbCtx->BCtxToBundle   0x00000000
  &pPhbCtx->BCtxStats     0x01707ff8
  pPhbCtx->BCtxConfig.slot/port/vcd     0/0/0
  pPhbCtx->BCtxConfig.inactivity        0
  pPhbCtx->BCtxConfig.ChannelIsSingle   0
  pPhbCtx->BCtxConfig.ChannelRank       0
  pPhbCtx->BCtxConfig.ChannelInterleave 0
  pPhbCtx->BCtxConfig.ChannelFragmentSz 0
 ======= pPhbCtx 0x01708084 iChannel = 1
  pPhbCtx->BCtxApiState   0
  pPhbCtx->BCtxB1B2       1
  pPhbCtx->BCtxMagic      0xdead7654
  pPhbCtx->BctxHandle     0x01708084
  pPhbCtx->BctxFastHandle 0x00000000
  pPhbCtx->BCtxConfig     0x0307c828
  pPhbCtx->BCtxDevCtx     0x01707f40
  pPhbCtx->BType          3
  pPhbCtx->BCtxToBundle   0x00000000
  &pPhbCtx->BCtxStats     0x017080f8
  pPhbCtx->BCtxConfig.slot/port/vcd     0/0/0
  pPhbCtx->BCtxConfig.inactivity        0
  pPhbCtx->BCtxConfig.ChannelIsSingle   0
  pPhbCtx->BCtxConfig.ChannelRank       0
  pPhbCtx->BCtxConfig.ChannelInterleave 0
  pPhbCtx->BCtxConfig.ChannelFragmentSz 0
. . .
 ======= pPhbCtx 0x01708d84 iChannel = 14
  pPhbCtx->BCtxApiState   0
  pPhbCtx->BCtxB1B2       14
  pPhbCtx->BCtxMagic      0xdead7654
  pPhbCtx->BctxHandle     0x01708d84
  pPhbCtx->BctxFastHandle 0x00000000
  pPhbCtx->BCtxConfig     0x0307c828
  pPhbCtx->BCtxDevCtx     0x01707f40
  pPhbCtx->BType          3
  pPhbCtx->BCtxToBundle   0x00000000
  &pPhbCtx->BCtxStats     0x01708df8
  pPhbCtx->BCtxConfig.slot/port/vcd     0/0/0
  pPhbCtx->BCtxConfig.inactivity        0
  pPhbCtx->BCtxConfig.ChannelIsSingle   0
  pPhbCtx->BCtxConfig.ChannelRank       0
  pPhbCtx->BCtxConfig.ChannelInterleave 0
  pPhbCtx->BCtxConfig.ChannelFragmentSz 0
 ======= pPhbCtx 0x01708f84 iChannel = 16
  pPhbCtx->BCtxApiState   0
  pPhbCtx->BCtxB1B2       16
  pPhbCtx->BCtxMagic      0xdead7654
  pPhbCtx->BctxHandle     0x01708f84
  pPhbCtx->BctxFastHandle 0x00000000
  pPhbCtx->BCtxConfig     0x0307c828
  pPhbCtx->BCtxDevCtx     0x01707f40
  pPhbCtx->BType          3
  pPhbCtx->BCtxToBundle   0x00000000
  &pPhbCtx->BCtxStats     0x01708ff8
  pPhbCtx->BCtxConfig.slot/port/vcd     0/0/0
  pPhbCtx->BCtxConfig.inactivity        0
  pPhbCtx->BCtxConfig.ChannelIsSingle   0
  pPhbCtx->BCtxConfig.ChannelRank       0
  pPhbCtx->BCtxConfig.ChannelInterleave 0
  pPhbCtx->BCtxConfig.ChannelFragmentSz 0
. . .
======= pPhbCtx 0x01709d84 iChannel = 30
  pPhbCtx->BCtxApiState   0
  pPhbCtx->BCtxB1B2       30
  pPhbCtx->BCtxMagic      0xdead7654
  pPhbCtx->BctxHandle     0x01709d84
  pPhbCtx->BctxFastHandle 0x00000000
  pPhbCtx->BCtxConfig     0x0307c828
  pPhbCtx->BCtxDevCtx     0x01707f40
  pPhbCtx->BType          3
  pPhbCtx->BCtxToBundle   0x00000000
  &pPhbCtx->BCtxStats     0x01709df8
  pPhbCtx->BCtxConfig.slot/port/vcd     0/0/0
  pPhbCtx->BCtxConfig.inactivity        0
  pPhbCtx->BCtxConfig.ChannelIsSingle   0
  pPhbCtx->BCtxConfig.ChannelRank       0
  pPhbCtx->BCtxConfig.ChannelInterleave 0
  pPhbCtx->BCtxConfig.ChannelFragmentSz 0
```

To display statistics for MLPPP over PRI B channel sequencer, use:

```
CLI> show system statistics pri sequencer
*************** Rx-reorder Configuration **************
 MLPPPoB Header mode               : LONG
 RMax                              : [50]
 Max Polling                       : [15]
 Late Packet Politic               : DROP
*******************************************************
*************** Rx-reorder Statistics *****************
---------------------- Treated ------------------------
Nb Packets forwarded in sequence         : [       129]
Nb Packets Dropped (because Late)       : [         0]
Nb LCP Packets Bundle (Dropped)         : [         0]
Nb Packets Dropped at Ring init         : [         0]
Nb Packets Dropped (internal Error)     : [         0]
Nb Late Packets Dropped (internal Error): [         0]
Nb Packets declared loss                : [         0]
Nb Packets Mean by job                  : [         1]
Nb job done                             : [       129]
------------------------ Rx ----------------------------
Nb Packets Rx From Drivers         : [       275]
   - SIG PPP Packets               +>: [       144]
   - Sequenced packets             +>: [       131]
     - IPCP Packets                | +>: [         2]
     - Data packets MLPPPoB        | +>: [       129]
       - Consider as LATE Packets  |   +>: [         0]
       - Sequenced Packets         |   +>: [       129]
       - Ring Full ( Dropped )     |   +>: [         0]
   - Dropped packets               +>: [         0]
     - Stack is NULL                 +>: [         0]
     - Mblk is NULL                  +>: [         0]
     - Channel is Down               +>: [         0]
-------------------------------------------------------
***************** Rx-reorder Status *******************
Nb Packet in the Ring              : [0]
Next expected MLPPP sequence nb    : [132]
*******************************************************
```

To reset statistics for MLPPP over PRI B channel sequencer, use:

```
CLI> show system statistics pri sequencer reset
```

7.5 ISDN DEBUG AND TRACES

You can place a fake ISDN call to verify the ISDN connection. The next command calls the given ISDN number with an ‘Unrestricted Data’ Bearer Capability (meaning it is a data call not a voice call). If the call is setup, it means the called party accepts data call and has eventually authenticated the calling number. After successful call setup, the call is then interrupted.

```
CLI> isdn test call <e164-nbr>
```

It is possible to display all the ISDN protocol messages exchanged between the OneOS-based router and the ISDN terminal. Such traces rely on the generic logging function (refer to "System Logging" section in "Traces and Events" chapter of "OneOS – Admin User Guide" document).

Enter the following command:

```
CLI> [no] debug isdn { 5/<0..7> | 2/0 | all } [layer { 1 | 2 | 3 | 1&2
                                                 | 2&3 | 1to3} [full] ]
```

The first command argument is the ISDN port(s), whose ISDN messages will be logged. The second argument `layer` is the monitored ISDN layers (1: physical, 2: LAPD, 3: call control and other ISDN applicative messages). The last optional argument `full` allows layer 3 information elements (IE) being decoded.

To display PPP/MLPPP traces:

```
CLI> [no] debug ppp
```

To display all WAN traces:

```
CLI> event filter add wan all show
```

To deactivate the WAN traces:

```
CLI> event filter remove all
```

7.6 ISDN CONFIGURATION EXAMPLES

Example of ISDN configuration:

```
configure terminal
interface bri 2/0
inactivity b-channel 5
activity hdlc timer 6
activity hdlc count 10
isdn
exit
no shutdown
execute
exit
```

Example of a Dial-in access:

```
configure terminal
interface isdn-dialer 2/0.1
incoming call management-site
```

```
address 0102030405060708
encapsulation ppp
profile 0
exit
execute
exit
```

Example of ATM interface backup:

```
configure terminal
interface atm 0
backup bri 2/0.6
address 384000
encapsulation ppp
profile 0
execute
exit
exit

virtual-template ppp 0
ipcp static
ip address 20.0.0.1
mru local 1500 no
mru remote 1500 no
magic number not-negotiable
authentication no
execute
exit
```

Example of MLPPP Connection over ISDN S0 (Outgoing Call):

```
configure terminal
interface bri 2/0
no shutdown
execute
exit

virtual-template ppp 5
ipcp static
ip address 20.0.0.4
mru local 1500 no
mru remote 1500 no
magic number not-negotiable
authentication no
reconnection datagram
execute
exit
interface isdn-dialer 2/0.5
outgoing call
mlppp 0
connection static
descriminator 1 toto
max-link 2
max-rcvunit 4000
header long
exit
address 2022024625
encapsulation ppp
profile 5
exit
execute
exit
exit
```

Example of MLPPP Connection over ISDN S0 (Incoming Call):

```
configure terminal
```

```
interface bri 2/0
no shutdown
execute
exit

virtual-template ppp 1
ipcp static
ip address 20.0.0.1
mru local 1500 no
mru remote 1500 no
authentication no
execute
exit

interface isdn-dialer 2/0.5
incoming call
address 1012024612
mlppp 0
connection static
max-link 2
max-rcvunit 4000
header long
exit
encapsulation ppp
profile 1
exit
execute
exit
exit
```

Callback example:

```
configure terminal
interface isdn-dialer 5/0.1
 inactivity b-channel 0
 both-way call
 address-outgoing 2123456789
 address-outgoing 1123456789 secondary
 callback-address 2123456789
 profile 1
 call-retry 2
  mlppp 0
   connection static-boda
   discriminator 1 ZWRS
   header long
   max-link 16
  exit
 exit
 execute
 dialer-group idle-timeout 20
 service-policy output LFIS
exit
interface bri 5/0
 isdn
  application-interface data
 exit
 no shutdown
exit
```

8 CONFIGURATION OF PSTN MODEM

8.1 PSTN MODEM SERVICE DESCRIPTION

The ONE60/100/200 products can be equipped with an optional analog modem. The modem enables a dial-up connection for backup of IP services.

The same backup types as for ISDN are provided:

1) Multipath backup routing: 2 routes must be configured. The main route has a metric that is lower than the metric going through the PSTN modem interface. When the main route is not available, the router sends packets through the analog modem interface. The PPP connection over the PSTN is set-up dynamically and the connection is released after an inactivity period. This inactivity period is due to the fact that there is either no traffic or the traffic is routed on the main route again (main link has become up again).

2) Interface backup: the serial interface/ATM PVC has configured so that the modem connection is set-up dynamically when the main interface is down. The modem connection is released when the main interface is up again.

8.2 PSTN MODEM CONFIGURATION

The configuration steps are the following:

1. Enable the modem interface and optionally set physical parameters.
2. Select whether the interface shall be using multipath or interface backup routing.
3. Set modem link parameters. These parameters are common to both backup modes.

8.2.1 Physical Interface Parameters

To enable the PSTN modem, enter in PSTN configuration mode from the `'configuration terminal'`:

```
CLI(configure)> [no] interface pstn 2/0
```

You may wish to change some parameters of the physical modem layer. To do so, enter in physical layer configuration as follows:

```
CLI(config-if)> physical-layer
```

The maximum physical-layer transmit unit (MTU) can be set in bytes (default: 1500) as follows:

```
CLI(physical-layer)> mtu <1..1500>
```

By default, the modem is disconnected even if there is no traffic. To configure the inactivity period, use the following command to set the inactivity timer (in seconds) after which the modem is disconnected (0 means no inactivity timer):

```
CLI(physical-layer)> channel inactivity timer <0..3600>
```

You can change the dialing type (default: tone dialing):

```
CLI(physical-layer)> modem { pulse-dial | tone-dial }
```

The country can be changed (default: France):

```
CLI(physical-layer)> country { FRANCE | GERMANY | ITALY | BELGIUM |
UNITED-KINGDOM | SPAIN }
```

It is possible to configure the ringing burst number received before answering an incoming call (default: 1 ring). The parameter is set under physical PSTN interface. It is used by the PSTN incoming call service as far as it is created (i.e. as far as the modem is set to accept the call).

To modify the value, the incoming call service must be removed and re-created.

```
CLI(physical-layer)> modem ring-nbr-to-answer <n>
```

To finish the interface configuration, enter the following commands:

```
CLI(physical-layer)> exit
CLI(config-if)> execute
CLI(config-if)> exit
CLI(configure)>
```

8.2.2 Interface Backup Routing

First select the interface, which must be backed up:

```
CLI(configure)> interface { atm 0.<id> | serial 0.<id> }
CLI(config-if)>
```

Then enable the PSTN backup:

```
CLI(config-if)> [no] backup pstn 2/0.<id>
```

`id` is an arbitrary number ranging from 1 to 32.

The next parameters to be configured are in 8.2.4 Common Link Parameters.

8.2.3 Multipath Backup Routing

From the global configuration terminal, enter the following commands:

```
CLI(configure)> interface pstn 2/0.<id>
CLI(config-if)> [no] outgoing-call [<name>]
CLI(out-call)>
```

`id` is an arbitrary number ranging from 1 to 32.

The next parameters to be configured are in 8.2.4 Common Link Parameters.

Do not forget that a route with an appropriate metric must exist to allow multi-path routing.

8.2.4 Common Link Parameters

To ease the reading of the configuration, you can enter an interface name as follows:

```
CLI(out-call)> sub-interface <name>
```

The called address must be entered:

```
CLI(out-call)> address <E.164>
```

By default, the encapsulation is PPP and is set as PPP as follows:

```
CLI(out-call)> encapsulation PPP
```

To apply PPP parameters to this connection, a profile number is attached. The profile corresponds to a virtual PPP template, which must have been previously defined. To attach the profile number (default: 0), use the following command:

```
CLI(out-call)> profile <0..32>
```

Some modem parameters can be set when entering in modem configuration mode:

```
CLI(out-call)> modem
```

The modem modulation can be forced:

```
CLI(out-call-modem)> modulation { V90 | NEGOTIATE | V22B }
```

Default: `Negotiate`. The modem compression can be selected:

```
CLI(out-call-modem)> compression { NONE | V42B | V44 }
```

Default: `V44`. The error correction scheme might be forced:

```
CLI(out-call-modem)> error-correction { NONE | V42 | MNP | LAPM}
```

Default: **`V42`**. The modem fallback can be disabled (default: enabled). The modem fallback permits the use of more robust modulation in case of poor quality line. The command is:

```
CLI(out-call-modem)> fallback { enable | disable }
```

You can also customize the Hayes command:

```
CLI(out-call-modem)> [no] customization <at-command-string>
```

Then enter 2 times `exit`, then the `execute` command and `exit` again.

```
CLI(out-call-modem)> exit
CLI(out-call)> exit
CLI(config-if)> execute
CLI(config-if)> exit
CLI(configure)>
```

8.3 PSTN MODEM STATISTICS AND TRACES

You can place a fake call to verify the connection. The next command calls the given number. If the call is setup, it means the called party has synchronized the modem. After successful call setup, the call is immediately interrupted.

```
CLI> pstn test call <e164-nbr> <slot>/<port>.<sub-interface>
```

To monitor the statistics of the PSTN connection:

```
CLI> show statistics pstn ppp
```

To monitor the statistics of the PSTN backup service:

```
CLI> show statistics pstn service
```

To obtain traces of the PSTN dial-up connection:

```
CLI> trace filter add wan dial pstn 1 show
```

As of V3.7R11 software release the above command is deprecated and is replaced by:

```
CLI> debug pstn 2
CLI> debug dialup 2
```

8.4 PSTN MODEM CONFIGURATION EXAMPLE

A router makes an outgoing call that backs up a main connection over ADSL. The ADSL connection is identified by the ATM PVC interface 0.1.

```
! PSTN interface declaration
! Enable shutdown of the connection after
! 20 s of inactivity
interface pstn 2/0
physical-layer
channel inactivty timer 20
exit
execute
exit

! PPP virtual template for the PSTN connection
virtual-template ppp 1
ipcp dynamic
authentication chap/pap
username me
password ******
reconnection datagram
execute
exit

interface atm 0.1
! Enter here the interface parameters
! like 'pvc pppoa ...'
backup pstn 2/0.1
address 1234
profile 1
execute
exit
exit

! Declare default routes then backup routes
ip route 0.0.0.0 0.0.0.0 atm 0.1
ip route 0.0.0.0 0.0.0.0 pstn 2/0.1 2
```

9 INTERNAL EDGE/GPRS MODEM

This section applies to the products ONECell25, ONECell35 (working as EDGE/GPRS modem) and ONE20/100 (equipped with an internal EDGE/GPRS module).

9.1 EDGE/GPRS MODEM PIN CODE

The internal modem requires a SIM card to be present to operate. At first modem installation, the PIN code must be installed. For security sake, the SIM code is not shown by the `show running-config` command. The PIN code is saved in encrypted form in a special file of the router flash file system.

The OneOS-based router gives the PIN code once; if the PIN code is incorrect, the product expects that the user enters another PIN code. If the user entered a series of (usually 3) wrong PIN codes, the SIM card is blocked by the GSM network. In that case, the user must contact the GSM provider to get a Pin code Unblocking Key (PUK).

The following commands are available from the CLI root. First, make sure that the SIM card is properly installed and detected by the system.

```
CLI> show edge-modem equipment
Edge Modem Information
 Manufacturer identification          :  Enfora, Inc.
 Equipment information                :  Enabler-II E Modem
 Revision identification              :  S/W v0.2.1
 Equipment information (IMEI)         :  01065600070006
 Firmware Package Version             :  44
 PIN status                           :  PIN code is not blocked (counter=3)
 SIM card status                      :  SIM card is present
 GPRS and EGPRS classes               :  10,10
```

Counter=3 indicates the number of remaining attempts to enter a correct PIN code.

PIN code installation of (an unblocked) SIM card is done with the next command:

```
CLI> edge-modem pin-code install <pin-code> <confirm-pin-code>
```

If the PIN code was successfully installed once, it can be changed as follows:

```
CLI> edge-modem pin-code change <old-pin-code> <pin-code> <confirm-pin-code>
```

To unblock a SIM card, ask the GSM provider for a temporary PIN code (PUK, usually an 8-digit PIN code) and specify the new PIN code:

```
CLI> edge-modem pin-code unblock <puk> <pin-code> <confirm-pin-code>
```

9.2 EDGE/GPRS CONNECTION CONFIGURATION

Foreword: there is no parameter to select EDGE or GPRS modulations. The network forces the modem to behave in the mode authorized by the network. Upstream and downstream bandwidth may not be steady as it depends on radio link quality and bandwidth allocation by the network. As the modem type is multi-class slot 10, the peak download throughput is 236 kbps and peak upload is 118 kbps.

The GSM provider usually gives the following parameters to setup GPRS/EDGE connection: an APN (Access Point Name), a username and password. This section explains where these parameters are entered.

First, configure a PPP virtual-template (see 5 for more information). Some parameters MUST have certain value to operate the EDGE/GPRS connection:

- Authentication: PAP (use the username and password provided by your GPRS network operator).
- Magic number: not-negotiated

```
CLI(configure)> virtual-template ppp <1-32>
! next command is optional: to learn DNS server from network
CLI(config-virt-ppp)> ipcp dns-accept
CLI(config-virt-ppp)> authentication pap
CLI(config-virt-ppp)> username <username>
CLI(config-virt-ppp)> password <password>
CLI(config-virt-ppp)> magic number not-negotiable
CLI(config-virt-ppp)> reconnection { automatic | datagram }
CLI(config-virt-ppp)> execute
CLI(config-virt-ppp)> exit
CLI(configure)>
```

Sometimes, the username and password parameters are empty. To enter an empty username / password, use the command:

```
CLI(config-virt-ppp)> authentication no
CLI(config-virt-ppp)> username “”
CLI(config-virt-ppp)> password “”
```

If `reconnection datagram` is selected, the connection is established only if IP packets have been routed to that interface. In case of link inactivity, the EDGE connection is disconnected. This mode is managed by the dialer-group function. If `reconnection automatic` is selected, the EDGE interface is permanently connected (and reconnected automatically in case of loss of connectivity). The EDGE interface can be forced down as long as another interface or IP route is up. This mode is managed by the dialer-watch function. To learn more about dialer-group and dialer-watch, consult the section 13.

The registration is first done at the GSM level then at the GPRS/EDGE level. Two registration failures are taken into account: lack of registration at start-up (power-up or reset) and lost of the registration (after a successful registration). For both cases a command line allows to set the time after witch a reset command (AT$RESET) is sent to the modem. An optional parameter allows choosing one registration or the other or both.

To reset the modem after some minutes of lack of registration at start-up use the following command. The default behavior is `power-on-registration reset after 15 GSM-or-GPRS`. Use `default power-on-registration reset` to return to the default behavior.

```
CLI(out-call-edge)> [no] power-on-registration reset after <minutes>
                                  [GSM-only | GPRS-only | GSM-or-GPRS]
```

To reset the modem after some minutes of lost of registration use the following command. The default behavior is `loss-of-registration reset after 5 GSM-or-GPRS`. Use `default loss-of-registration reset` to return to the default behavior.

```
CLI(out-call-edge)> [no] loss-of-registration reset after <minutes>
```

```
                                              [GSM-only | GPRS-only | GSM-or-GPRS]
```

To avoid a modem lock when there is no traffic at all, it is possible to disconnect the call when there is no traffic either in both directions or only in the receiving direction. For both cases a command line allows to set the time after witch the call is disconnected. It will be reconnected depending on the reconnection parameter of the virtual-template PPP.

To disconnect the call after some minutes without traffic in both directions use the following command. The default behavior is `disconnect-timeout on-no-traffic after 240` (4 hours).

```
CLI(out-call)> [no] disconnect-timeout on-no-traffic after <minutes>
```

To disconnect the call after some minutes without response traffic use the following command. The default behavior is '`disconnect-timeout on-no-response after 5`'.

```
CLI(out-call)> [no] disconnect-timeout on-no-response after <minutes>
```

To configure the logical IP interface of the modem, create a PSTN interface:

```
CLI(configure)> interface pstn 2/0.<1-255>
CLI(config-if)> outgoing-call
! <gprs-service-nbr> is usually *99***1#
CLI(out-call)> address <gprs-service-nbr>
! PPP virtual template
CLI(out-call)> profile <virt-ppp-profile>
CLI(out-call)> disconnect-timeout on-no-traffic after <minutes>
CLI(out-call)> disconnect-timeout on-no-response after <minutes>
CLI(out-call)> edge-modem
CLI(out-call-edge)> access-point-name <apn>
CLI(out-call-edge)> power-on-registration reset after <min> GSM-or-GPRS
CLI(out-call-edge)> loss-of-registration reset after <minutes>
CLI(out-call-edge)> exit
CLI(out-call)> exit
CLI(config-if)> execute
CLI(config-if)> exit
```

9.3 EDGE/GPRS CONFIGURATION EXAMPLES

Simple Always-On Internet Connection over EDGE:

```
interface bvi 1
 ip address 192.168.1.1 255.255.255.0
 bridge-group 1
exit
interface fastEthernet 0/0
 no ip address
 bridge-group 1
exit
interface fastEthernet 1/0
 no ip address
 bridge-group 1
exit
virtual-template ppp 1
 ipcp dns-accept
 authentication pap
 username orange
 password orange
 magic number not-negotiable
 reconnection automatic
 execute
exit
```

```
interface pstn 2/0.1
 outgoing-call
  address *99***1#
  profile 1
exit
execute
 ip nat inside overload
exit
ip route 0.0.0.0 0.0.0.0 Pstn 2/0.1
ip dns-proxy dns-server learn
ip name-server 127.0.0.1
ip dhcp pool CLIENT_LAN
 network range 192.168.1.2 192.168.1.254
 default-router 192.168.1.1
 dns-server 192.168.1.1
exit
exit
```

EDGE Interface started only if an interface is down

```
interface dialer 0
! PPPoE configuration
!...
exit
dialer watch-list watchpppoe
 interface dialer 0
exit
virtual-template ppp 1
 ipcp dns-accept
 authentication pap
 username orange
 password orange
 magic number not-negotiable
 reconnection automatic
 execute
exit
interface pstn 2/0.1
 outgoing-call
  address *99***1#
  profile 1
exit
 execute
 ip nat inside overload
 dialer watch-group watchpppoe
exit
ip route 0.0.0.0 0.0.0.0 dialer 0
ip route 0.0.0.0 0.0.0.0 Pstn 2/0.1 200
```

9.4 EDGE/GPRS STATISTICS AND TROUBLESHOOTING

To show information about the EDGE module:

```
CLI> show edge-modem equipment
```

To display information about the radio network:

```
CLI> show edge-modem network
```

To display the AT commands passed by the router to modem:

```
CLI> debug dialup <level: 1-4>
```

10 INTERNAL UMTS MODEM

This section applies to the product ONECell35.

Note: most of the commands are similar to the ones of the internal EDGE/GPRS modem of the ONECell25 product. Refer to them if more explanations are needed (mind to replace the `edge-modem` ***command by the*** `cellular-radio` ***command).***

10.1 UMTS MODEM PIN CODE

The internal modem requires a SIM card to be present to operate. At first modem installation, the PIN code must be installed. For security sake, the SIM code is not shown by the `show running-config` command. The PIN code is saved in encrypted form in a special file of the router flash file system.

The ONECell35 gives the PIN code once; if the PIN code is incorrect, the product expects that the user enters another PIN code. If the user entered a series of (usually 3) wrong PIN codes, the SIM card is blocked by the UMTS network. In that case, the user must contact the UMTS provider to get a Pin code Unblocking Key (PUK).

The following commands are available from the CLI root. First, make sure that the SIM card is properly installed and detected by the system.

```
CLI> show cellular-radio equipment
 Cellular Radio Modem Information
  Manufacturer identification           :  Sierra Wireless, Inc.
  Equipment information                 :  MC8785V
  Revision identification               :  J0_0_4_1AP …
  Equipment information (IMEI)          :  351532020030976

 SIM Card Information
  SIM card status                       :  SIM card is present
  SIM International Mobile Subscriber Identity IMSI  :  208101384000299
  Integrated Circuit Card ID            :  98330131705364604060

 PIN Information
  PIN code status                       :  entered OK
```

PIN code installation of (an unblocked) SIM card is done with the next command:

```
CLI> cellular-radio pin-code install <pin-code> <confirm-pin-code>
```

If the PIN code was successfully installed once, it can be changed as follows:

```
CLI> cellular-radio pin-code change <old-pin-code> <pin-code> <confirm-
                                                                pin-code>
```

The current PIN code can be uninstalled with the next command:

```
CLI> cellular-radio pin-code uninstall
```

To unblock a SIM card, ask the UMTS provider for a temporary PIN code (PUK, usually an 8-digit PIN code) and specify the new PIN code:

```
CLI> cellular-radio pin-code unblock <puk> <pin-code> <confirm-pin-code>
```

To have the SIM card not requesting the PIN code, use the following command:

```
CLI> cellular-radio pin-code disable <pin-code>
```

To return to the default behavior (entering PIN code is required), use the following command:

```
CLI> cellular-radio pin-code enable <pin-code>
```

10.2 UMTS CONNECTION CONFIGURATION

Foreword: the UMTS modem is also able to work as an EDGE/GPRS modem.

Upstream and downstream bandwidths may not be steady as they depend on radio link quality and bandwidth allocation by the network. In UMTS mode the peak download throughput is 7200 kbps and peak upload is 2000 kbps. In EDGE/GPRS mode the modem type is multi-class slot 12, the peak download throughput is 236 kbps and peak upload is 236 kbps.

The UMTS provider usually gives the following parameters to setup UMTS connection: an APN (Access Point Name), a username and password. This section explains where these parameters are entered.

First, configure a PPP virtual-template (see 5 for more information). Some parameters MUST have certain value to operate the UMTS connection:

- Authentication: PAP (use the username and password provided by your UMTS network operator).
- Magic number: not-negotiated

```
CLI(configure)> virtual-template ppp <1-32>
! next command is optional: to learn DNS server from network
CLI(config-virt-ppp)> ipcp dns-accept
CLI(config-virt-ppp)> authentication pap
CLI(config-virt-ppp)> username <username>
CLI(config-virt-ppp)> password <password>
CLI(config-virt-ppp)> magic number not-negotiable
CLI(config-virt-ppp)> reconnection { automatic | datagram }
CLI(config-virt-ppp)> execute
CLI(config-virt-ppp)> exit
CLI(configure)>
```

Sometimes, the username and password parameters are empty. To enter an empty username / password, use the command:

```
CLI(config-virt-ppp)> authentication no
CLI(config-virt-ppp)> username “”
CLI(config-virt-ppp)> password “”
```

If `reconnection datagram` is selected, the connection is established only if IP packets have been routed to that interface. In case of link inactivity, the UMTS connection is disconnected. This mode is managed by the dialer-group function. If `reconnection automatic` is selected, the UMTS interface is permanently connected (and reconnected automatically in case of loss of connectivity). The UMTS interface can be forced down as long as another interface or IP route is up. This mode is managed by the dialer-watch function. To learn more about dialer-group and dialer-watch, consult the section 13.

The UMTS modem is able to use the 3G and/or the 2G radio access technologies. Use the following command to select the radio access technology (auto is the default value):

```
CLI(out-call-cellular)> radio-access-technology {2G-only | 3G-only |auto}
```

Two registration failures are taken into account: lack of registration at start-up (power-up or reset) and lost of the registration (after a successful registration). For both cases an attach command (`AT!CGATT`) is immediately sent to the modem to register again. Only if the re-register does not succeed, a reset command (`AT!RESET`) is sent to the modem. A command line allows setting the time after witch the reset command is sent.

To reset the modem after some minutes of lack of registration at start-up use the following command. The default behavior is `power-on-registration reset after 15`. Use `default power-on-registration reset` to return to the default behavior.

```
CLI(out-call-cellular)> [no] power-on-registration reset after <minutes>
```

To reset the modem after some minutes of lost of registration use the following command. The default behavior is `loss-of-registration reset after 5`. Use `default loss-of-registration reset` to return to the default behavior.

```
CLI(out-call-cellular)> [no] loss-of-registration reset after <minutes>
```

To avoid a modem lock when there is no traffic at all, it is possible to disconnect the call when there is no traffic either in both directions or only in the receiving direction. For both cases a command line allows to set the time after witch the call is disconnected. It will be reconnected depending on the reconnection parameter of the virtual-template PPP.

To disconnect the call after some minutes without traffic in both directions use the following command. The default behavior is `disconnect-timeout on-no-traffic after 240` (4 hours).

```
CLI(out-call)> [no] disconnect-timeout on-no-traffic after <minutes>
```

To disconnect the call after some minutes without response traffic use the following command. The default behavior is `disconnect-timeout on-no-response after 5`.

```
CLI(out-call)> [no] disconnect-timeout on-no-response after <minutes>
```

To configure the logical IP interface of the modem, create a PSTN interface:

```
CLI(configure)> interface pstn 2/0.<1-255>
CLI(config-if)> outgoing-call
! <umts-service-nbr> is usually *99#
CLI(out-call)> address <umts-service-nbr>
! PPP virtual template
CLI(out-call)> profile <virt-ppp-profile>
CLI(out-call)> disconnect-timeout on-no-traffic after <minutes>
CLI(out-call)> disconnect-timeout on-no-response after <minutes>
CLI(out-call)> cellular-radio
CLI(out-call-cellular)> access-point-name <apn>
CLI(out-call-cellular)> radio-access-technology <rat>
CLI(out-call-cellular)> exit
CLI(out-call)> exit
CLI(config-if)> execute
CLI(config-if)> exit
```

10.3 UMTS MODEM CONFIGURATION EXAMPLES

Simple Always-On Internet Connection over UMTS

```
interface bvi 1
 ip address 192.168.1.1 255.255.255.0
 bridge-group 1
exit
interface fastEthernet 0/0
 no ip address
 bridge-group 1
exit
interface fastEthernet 1/0
```

```
 no ip address
 bridge-group 1
exit
virtual-template ppp 1
 ipcp dns-accept
 authentication pap
 username orange
 password orange
 magic number not-negotiable
 reconnection automatic
 execute
exit
interface pstn 2/0.1
 outgoing-call
  address *99#
  profile 1
exit
execute
 ip nat inside overload
exit
ip route 0.0.0.0 0.0.0.0 Pstn 2/0.1
ip dns-proxy dns-server learn
ip name-server 127.0.0.1
ip dhcp pool CLIENT_LAN
 network range 192.168.1.2 192.168.1.254
 default-router 192.168.1.1
 dns-server 192.168.1.1
exit
exit
```

UMTS Interface started only if an interface is down

```
interface dialer 0
! PPPoE configuration
!...
exit
dialer watch-list watchpppoe
 interface dialer 0
exit
virtual-template ppp 1
 ipcp dns-accept
 authentication pap
 username orange
 password orange
 magic number not-negotiable
 reconnection automatic
 execute
exit
interface pstn 2/0.1
 outgoing-call
  address *99#
  profile 1
exit
 execute
 ip nat inside overload
 dialer watch-group watchpppoe
exit
ip route 0.0.0.0 0.0.0.0 dialer 0
ip route 0.0.0.0 0.0.0.0 Pstn 2/0.1 200
```

10.4 UMTS MODEM STATISTICS AND TROUBLESHOOTING

To show information about the UMTS module (see result in 10.1):

```
CLI> show cellular-radio equipment
```

To display information about the radio network (see signal strength table below):

```
CLI> show cellular-radio network
Status :
 GSM  registration status                    :  registered (home network)
 GPRS registration status                    :  registered (home network)
 Current selected operator                   :  F SFR
 Signal strength                             :  bad (-83 dBm)
 Current radio access technology             :  HSDPA/HSUPA


Statistics :
 Successful GSM  registrations                         :  2
 Successful GPRS registrations                         :  1
 Loss of GSM  registration                             :  0
 Loss of GPRS registration because of loss of GSM      :  0
 Loss of GPRS registration because of network          :  0
 Reset on loss of GSM  registration                    :  0
 Reset on loss of GPRS registration                    :  0
 Reset on failed initial registration                  :  0
CLI>
```

Signal strength table: number of LED (quality) vs. signal level in dBm

Number of LED	Signal quality	Signal level in 2G mode	Signal level in 3G mode
1 LED is ON	bad	S < −95 dBm	S < −105 dBm
2 LEDs are ON	fair	−95 dBm ≤ S < −85 dBm	−105 dBm ≤ S < −99 dBm
3 LEDs are ON	good	−85 dBm ≤ S < −75 dBm	−99 dBm ≤ S < −89 dBm
4 LEDs are ON	excellent	−75 dBm ≤ S	−89 dBm ≤ S

To display the AT commands passed by the router to modem:

```
CLI> debug dialup <level: 1-4>
```

10.5 UMTS MODULE FIRMWARE DOWNLOAD

In same cases a different UMTS module firmware has to be used. Refer to the OneAccess Customer Support Department to get the files corresponding to an alternate firmware release.

To download a file in the UMTS module, use the following command in CLI root mode:

```
CLI> cellular-radio firmware-upgrade <file-path>
```

Note: It may append that the firmware release consists of two files, a boot file and an application file. In such a case, the boot file must be downloaded first.

11 EXTERNAL MODEM USING AT COMMANDS (GPRS)

This functionality is discontinued as of V4.2 software release.

12 AUTHENTICATION OF DSL LINK BASED ON PSTN/ISDN

Opening PPP over a DSL uplink usually takes place when the access router is authenticated using a PAP or CHAP-based password authentication. The authentication is done by the router/DSLAM terminating the PPP session. However, this piece of equipment does not contain the database of users and passwords. This database is hosted (or simply queried) by a TACACS+ or RADIUS server. The following security issue is raised in the following case: if the terminating PPP router does not have information about the copper line, PPP is authenticated only based on the login/password pair. This means that the same router can be plugged to any DSL line and get network access.

On a PSTN/ISDN connection, PPP is authenticated after login/password authentication AND identification of the ISDN/PSTN phone number associated with the copper line.

Authentication is an OneAccess patent-pending technique which goal is to acquire permission to enable the uplink interface when the router has just rebooted. Authentication uses standard PPP features to establish a session on a PSTN/ISDN interface. Authentication is successful when the PPP session has been established. This PPP session is then either closed or re-used by Autoconfiguration feature.

Authentication success allows the uplink interface(s) to be enabled and PSTN/ISDN backup (dial-out) is allowed to be activated. Authentication failure forbids the use of the uplink interface(s) and PSTN/ISDN backup (dial-out) is not allowed to be activated.

PSTN/ISDN incoming calls are not concerned by authentication. PSTN/ISDN dial-in is possible at any time. Note that an active dial-in PSTN/ISDN connection prevents the authentication process. The user is able to configure the number of retries and the delay between each retries of every PSTN/ISDN authentication call: this is made using configuration parameters in the outgoing or both call description at PSTN/ISDN interface level.

12.1 PSTN/ISDN AUTHENTICATION CONFIGURATION

The configuration is realized in two steps: first, the authentication profile is configured; then, the authentication template is applied to interfaces that must not be enabled until authentication is successful.

The authentication profile is configured with the next commands:

```
CLI(configure)> authentication oappp <client-method-name>
CLI(config-auth)> enable aux-led
CLI(config-auth)> disable aux-led
CLI(config-auth)> enable hangup-immediate
CLI(config-auth)> disable hangup-immediate
CLI(config-auth)> client <client-name>
CLI(config-auth-client)> interface { isdn <slot>/<port>.<id>
                                   | pstn <slot>/<port>.<id> }
CLI(config-auth-client)> exit
CLI(config-auth)> execute
CLI(config-auth)> exit
```

The `aux-led` parameter allows to the front-panel auxiliary LED to watch the authentication state. The default value is `enable aux-led`. The `hangup-immediate` parameter allows to hang-up the PSTN/ISDN connection as soon as PPP reached the UP state. The default value is `enable hangup-immediate`. `client-method-name` is further referenced in the configuration as the service authentication profile. `client-name` is an arbitrary string that is provisioned for future use.

The authentication is done on the PSTN/ISDN sub-interface referenced by the `interface pstn` command. The PSTN/ISDN interface is a standard outgoing call PPP interface. Syntax example:

```
interface { pstn | isdn-dialer } <slot>/<port>.<id>
  { outgoing call | both-way call }
    address <E164>
    ! WARNING: secondary is not supported on ISDN interface
    address <E164> secondary
    call-retry <number_of_retry> [<number_of_seconds>]
    profile <ppp-virt-template-id>
    exit
   execute
  exit
```

Then, the authentication service can be applied to uplink interfaces:

```
interface atm <port>.<id>
  pvc { pppoa | ipoa | pppoeoa } vpi <vpi> vci <vci>
    service-authentication <client-method-name>
    execute
    exit
  exit
```

12.2 PSTN/ISDN AUTHENTICATION EXAMPLE

```
authentication oappp pstnauthentication
 client dialup
  interface pstn 2/0.1
  exit
 execute
exit
interface pstn 2/0.1
 outgoing call
  address 12345678
  call-retry 100
  profile 2
 exit
 execute
exit
interface atm 0.2
 pvc pppoa vpi 0 vci 32
  service-authentication pstnauthentication
  execute
  exit
exit
```

12.3 PSTN/ISDN AUTHENTICATION TRACES AND DEBUG

Autoconfiguration traces are activated with the following event command:

```
CLI> event filter add wan auth all show
```

To remove such traces:

```
CLI> event filter remove wan auth all show
```

The possible failure reasons are:

local failure: no response from modem	hardware failure
local failure: modem ERROR	software failure
remote failure: modem BUSY	remote side is busy
remote failure: modem NO CARRIER	failure to connect with remote side
remote failure: modem NO ANSWER	remote side don't answer
local failure: modem NO DIAL TONE	PSTN cable not connected
local failure: modem DELAYED	automatic dial out not yet successful
local failure: modem BLACKLISTED	remote “blacklisted” (too many unsuccessful attempts)

To display the current authentication status:

```
CLI> show authentication status
```

To display the authentication statistics:

```
CLI> show statistics authentication
```

To display the authentication configuration:

```
CLI> show configuration authentication
```

13 ACTIVITY CONTROL OF BACKUP & SWITCHED INTERFACES

13.1 INTRODUCTION TO SWITCHED INTERFACE

Non-switched interface are interfaces such ATM PVC or FastEthernet which have two states: up and down. Switched interfaces are interfaces that can be established on-demand (a circuit must be established first between a caller and a called piece of equipment before data can flow through the interface); therefore it has three states: down (circuit cannot be established), dormant (the circuit is not established, but ready for establishment), and up (circuit established). An ISDN "outgoing call" interface is a switched interface.

An ISDN "incoming call" interface is not a switched interface because only two states are available for this interface (up –connected–, down –waiting to be called–). An ISDN "both-way call" is a switched interface when it dials out (same as "outgoing call") and not switched when it is called (same as "incoming call").

On switched interfaces, a supervisor software module is responsible for opening and closing the circuit. The supervisor receives open/close requests from the dialer-group and from the dialer-watch-list modules. This supervisor also informs the dialer-group and the dialer-watch-list about the circuit status (down, opening, open, and closing).

Dialer-group permits to open a switched interface on-demand when packets match some criteria and close the circuit after a certain time of inactivity. Dialer-watch-list requests a circuit to be permanently established as long as certain criteria are not fulfilled. Because both software modules are different entities, if dialer-watch-list has required ISDN to be open, the dialer-group may request to close the ISDN communication.

13.2 DIALER-GROUPS

When configuring a dial-on-demand connection on a BRI interface, it is possible to apply dialer-groups, that is to say a control on outgoing and/or incoming data flow, in order to establish and maintain connection.

For example, an administrator may wish to establish an ISDN link, only when an outgoing telnet flow must be routed via the BRI interface. This control is made by an access-list that checks if the packet is a telnet packet or not. If the packet matches the access-list, the connection is established. Otherwise, the connection is not established. This type of filtering does not drop packets if the connection is already established.

Similarly, once the link is established, the administrator may want the connection to stay up as long as user traffic is routed onto this interface. A filter checks that the traffic maintaining the link activity is not a mandatory flow. For example, SNTP or RIP sends periodically packets. It is desirable to disconnect the ISDN link, when SNTP or RIP is the only flows using ISDN.

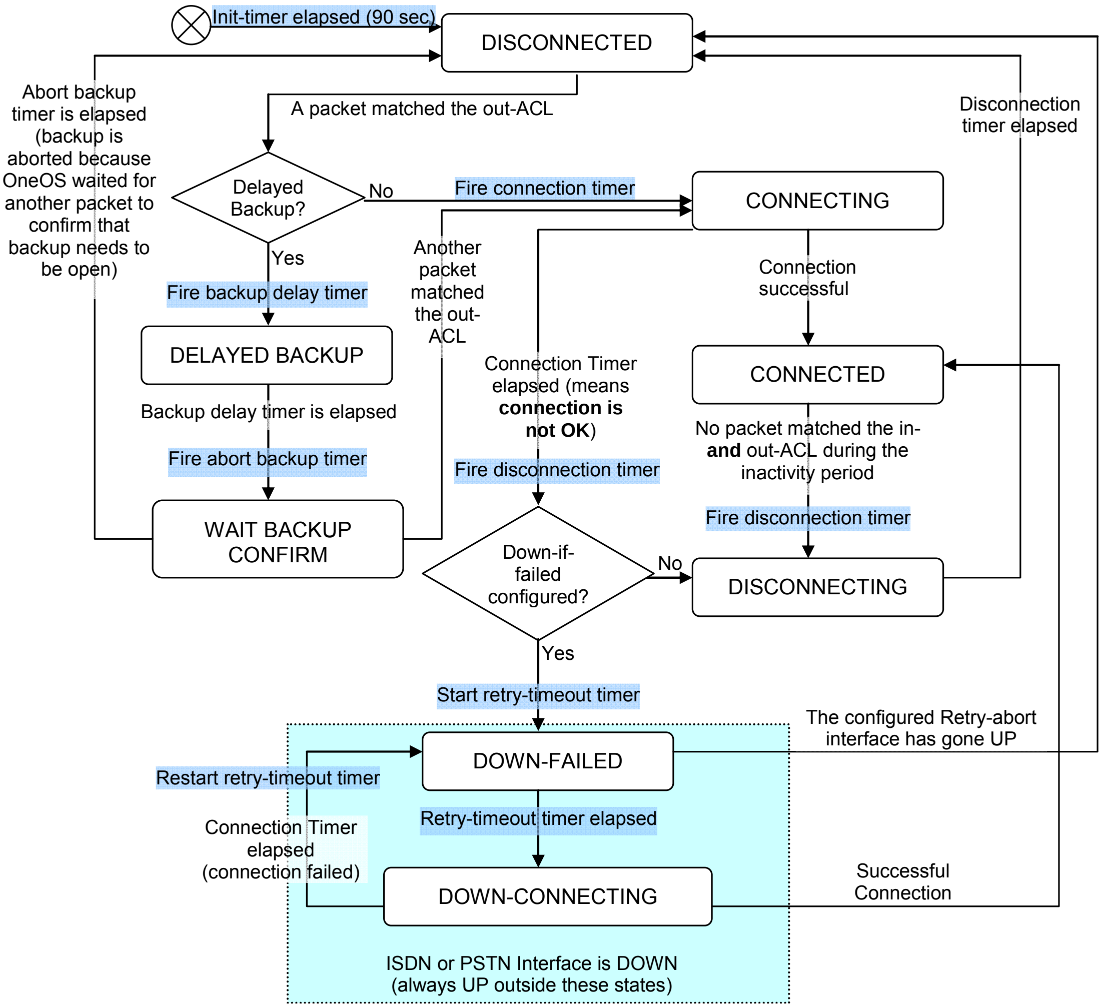
To maintain the connected state of the ISDN link opened via the dialer-group, an idle timer is reset each time a packet is matched by the access-list configured in the dialer-group. Non-critical flows must not match these access lists. **If no access-list is configured, every packet passing through the interface resets the inactivity timer. It can be then necessary to define an access-list in both directions to allow the interface to be closed via inactivity**.

Although using access lists, a dialer-group filters packets in a different place than traditional access lists.

The filtering is done as the first step for incoming traffic and as the last step for outgoing traffic. (NAT might be activated and yield to unexpected results when not paying attention to this fact.)

Sometimes, it is desirable not to establish the backup link immediately. For example, an ADSL BAS may force a PPP renegotiation by clearing the PPP session. Reestablishing PPP takes some time and it is not mandatory in that case to open the backup interface. To realize such behavior, two timers were implemented. They are a delayed backup timer and a delayed backup abort timer: when a packet is routed to the backup interface, the timers are fired and this packet as well as all subsequent packets is dropped until the delayed backup timer is expired. If a packet is routed to that interface before the abort timer expires, the backup interface is opened. If no packet is routed to the interface in that time frame, the interface is not opened.

The following diagram shows the dialer-group status. On the diagram below, you can notice that OneOS is ready to dial when a packet is routed to the ISDN interface, matched by the ACL and the dialer-group is in the IDLE state. **An explanation of the "DOWN-FAILED" status is provided after the diagram**.


Init-timer elapsed (90 sec)
DISCONNECTED
A packet matched the out-ACL
Abort backup timer is elapsed (backup is aborted because OneOS waited for another packet to confirm that backup needs to be open)
Delayed Backup?
No
Fire connection timer
CONNECTING
Yes
Fire backup delay timer
Another packet matched the out-ACL
Connection successful
Disconnection timer elapsed
DELAYED BACKUP
Backup delay timer is elapsed
Fire abort backup timer
Connection Timer elapsed (means connection is not OK)
Fire disconnection timer
CONNECTED
No packet matched the in- and out-ACL during the inactivity period
Fire disconnection timer
WAIT BACKUP CONFIRM
Down-if-failed configured?
No
DISCONNECTING
Yes
Start retry-timeout timer
The configured Retry-abort interface has gone UP
Restart retry-timeout timer
DOWN-FAILED
Retry-timeout timer elapsed
Connection Timer elapsed (connection failed)
DOWN-CONNECTING
Successful Connection
ISDN or PSTN Interface is DOWN (always UP outside these states)

If the option `down-if-failed` is not configured, the interface managed by the dialer-group is always seen as being UP by the IP layer (as displayed in `show interfaces isdn-dialer`). Because the IP interface is seen UP, the IP routes set on this interface are always valid. The reason is the following: the ISDN or PSTN interface can always serve as backup for the main interface. Even if the dial-out connection failed, maybe the next connection attempt will be successful. That is the reason why the ISDN or PSTN interface managed by the dialer-group stays UP for further dial-out attempts. However, it might be necessary that this ISDN/PSTN interface be backed up by another interface, for example:

- Backup of an ISDN sub-interface by another ISDN sub-interface (two different destinations with different PPP profiles)
- Backup of an ISDN interface by a PSTN or GPRS interface

If the option `down-if-failed` is not configured, both above-mentioned backup scenarios cannot be realized. That is why new states were introduced in the dialer-group state machine. The new states forces the interface to be DOWN if the connection failed. Subsequently the routes set on this interface disappear and routes with a lower metric/distance become then valid routes (and can redirect packets to a secondary backup path). While the interface is DOWN, it constantly tries to reconnect. The IP interface (as shown by `show interfaces isdn-dialer`) goes up again if one of the following conditions is met:

- Successful connection
- The retry-abort interface has gone UP. This interface (e.g.: `atm 0.1`) is usually the main interface. If this interface is UP, no backup is needed; therefore, ISDN/PSTN connection is aborted.

To configure an outbound dialer-group on a BRI interface, use the following command line:

```
CLI(configure)> interface { isdn-dialer | pstn } <slot>/<port>.<cnx-id>
CLI(config-if)> dialer-group ip <acl-out> out
```

`acl-out` is the defined access-list. It filters the flows that are allowed to open the connection. In general, such access list is made up of a "permit any" and deny commands for the protocols that must not maintain the connection open.

Inbound flow can also be checked against an inbound dialer-group, so as to maintain line established:

```
CLI(config-if)> dialer-group ip <acl-in> in
```

`acl-in` is the defined access-list. When a packet matches this access list, the idle timer is reset. So, the above command can be important to determine when the ISDN connection can be disconnected.

To remove an outbound dialer-group, use the following command line:

```
CLI(config-if)> no dialer-group ip out
```

To remove an inbound dialer-group, use the following command line:

```
CLI(config-if)> no dialer-group ip in
```

Note that if no dialer-groups are configured on a BRI interface, all traffic is authorized to connect and to maintain the ISDN line open.

To configure the inactivity timer on BRI interface, use the following command line in interface command mode:

```
CLI(config-if)> dialer-group idle-timeout <seconds>
```

Default value for inactivity timer is 90 seconds.

To configure the delayed backup, use the following command line in interface command mode:

```
CLI(config-if)> [no] dialer-group delayed-backup
CLI(config-if)> dialer-group delayed-backup delayed-period <seconds>
CLI(config-if)> dialer-group delayed-backup abort-timeout <seconds>
```

Default values: no delayed backup, delayed-period = 10 sec, abort-timeout = 10 sec.

The command `dialer-group differed-backup` … is deprecated.

The connection establishment timer is a timer that monitors the establishment of the interface (from open request until PPP is UP). By default, it is 30 sec for ISDN and 60 sec for PSTN. To modify this timer:

```
CLI(config-if)> dialer-group connection-wait-time <seconds>
```

The initialization timer forbids the establishment of PSTN/ISDN while another link is going up (e.g. a DSL line is synchronizing). Default: 120 seconds. To modify this timer:

```
CLI(config-if)> dialer-group initialization-timeout <seconds>
```

To force the interface to be DOWN if the connection failed, use the next command:

```
CLI(config-if)> dialer-group down-if-failed retry-timeout <seconds>
                        [retry-abort atm 0.<id>] [max-retry <nbr>]
```

Default: disabled. There are several connection attempts every (connection-wait-time + retry-timeout) seconds (warning: the ISDN sub-interface can be configured such that it dials different ISDN numbers and retries several times). The retry-timeout must be longer than the maximum time required for an ISDN interface to dial and fail all connection trials). If the retry-abort interface is UP, the automatic reconnection is disabled and the interface goes UP again (For example on how to use such option, see the second example in this section). If the max-retry is not null and after max-retry connection attempts, the automatic reconnection is disabled and the interface goes UP again. If max-retry is null and retry-abort is not set (default values), connection attempts are done until connection succeed.

To deactivate the `down-if-failed` option:

```
CLI(config-if)> no dialer-group down-if-failed
```

The following example shows how `acl_out` and `acl_in` access-list permit access only from a specific sub-network to another sub-network.

```
ip access-list extended acl_out
 permit ip 220.12.1.0 0.0.0.255 100.0.2.0 0.0.0.255
exit
ip access-list extended acl_in
 permit ip 100.0.2.0 0.0.0.255 220.12.1.0 0.0.0.255
exit
interface isdn-dialer 2/0.1
 dialer-group ip acl_out out
 dialer-group ip acl_in in
 dialer-group idle-timeout 30
exit
```

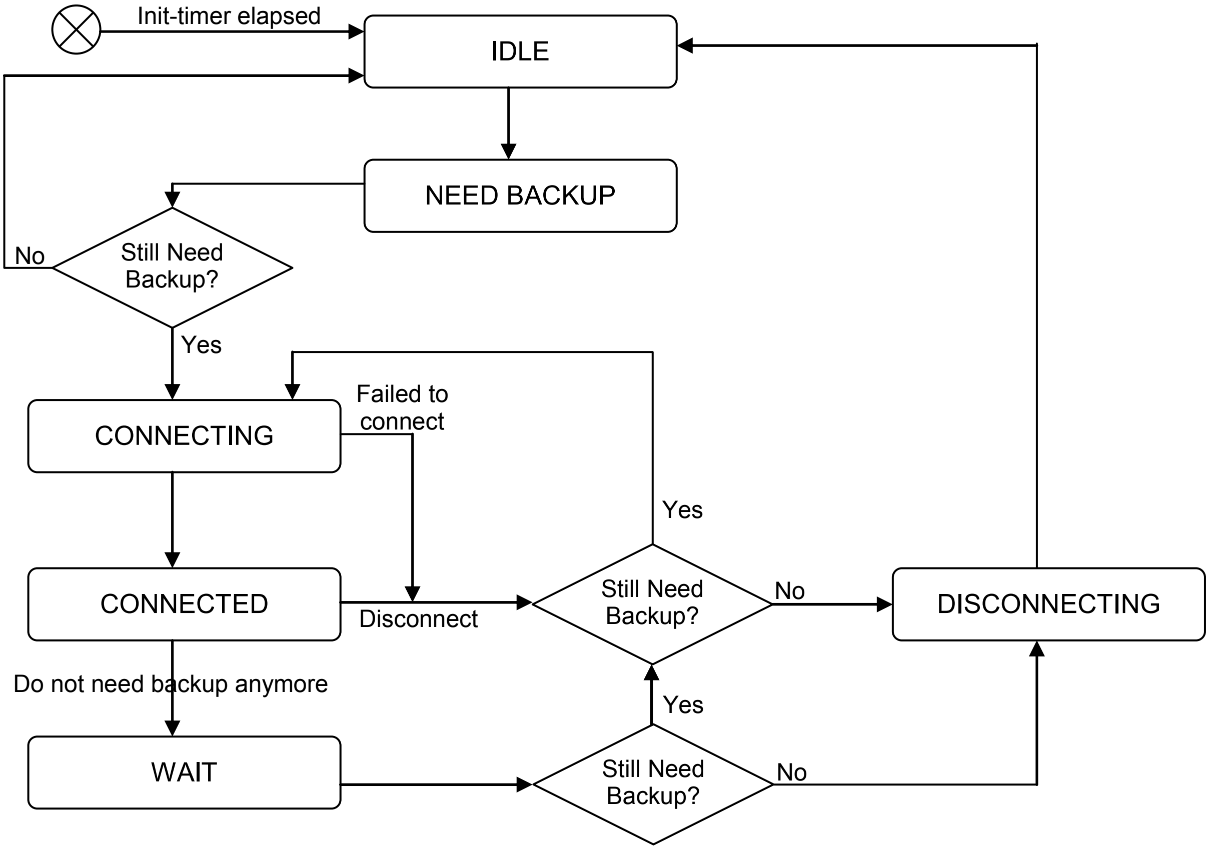
The next example activates the backup of an ISDN interface (`isdn 2/0.1`) by a PSTN interface (`pstn 0/0.1`), while ISDN is the backup of `atm 0.1` (backup priorities: 1) ATM, 2) ISDN and 3) PSTN):

```
! preferred path is ATM, then ISDN
ip route 0.0.0.0 0.0.0.0 atm 0.1
ip route 0.0.0.0 0.0.0.0 isdn 2/0.1 200
ip route 0.0.0.0 0.0.0.0 pstn 0/0.1 201
interface isdn 2/0.1
 dialer-group down-if-failed retry-timeout 60 retry-abort atm 0.1
exit
```

13.3 BACKUP USING ROUTE MONITORING (DIALER WATCH-LIST)

When using dialer-groups, you are able to control the opening and closing of switched interfaces (such as PSTN, BRI …) based on link activity for selected flows. The dialer watch-list meets another type of requirement: when the main path to a destination is no more available, a backup interface must be open even if there is no traffic. For that purpose, selected routes and/or interfaces are monitored. When one or more of the monitored routes disappear from the routing table, the backup interface is open and remains open as long as the missing routes have not appeared on an interface other than the backup interface.

The diagram below shows the state diagram of a dialer watch-list.



Configuration

Note: as of V4.3R2E2 software release, dialer watch-list is not supported outside default VRF.

First, you must configure a dialer watch-list to configure the monitored routes and interfaces. The following command is used under configuration terminal:

```
CLI(configure)> dialer watch-list [logical-any | logical-all] <list-name>
CLI(watch-routes)>
```

`list-name` identifies the watch list. The `logical-any` keyword (default) indicates that the backup can be activated when any interface or route declared in the watch-list is missing. When using `logical-all`, all elements declared in the list must be missing to activate the backup interface.

Then, you can declare the list of routes and interfaces to monitor:

```
CLI(watch-routes)> ip <address> <mask>
CLI(watch-routes)> interface <type> <unit>
```

At startup, routes and interfaces are not monitored for a certain time. **Therefore, backup is forbidden for a certain time at startup**. This allows time for the DSL interface to synchronize with the DSLAM and routing protocols to provide routes (especially those that must be monitored). The timer is configured as follows:

```
CLI(watch-routes)> initial-check-delay <seconds>
```

Default delay: 120 seconds. Once the list is configured, enter **`exit`**.

Then, the dialer watch-list is applied on the backup interface. Example, for a BRI (ISDN dialer) interface:

```
CLI(configure)> interface isdn-dialer <slot>/<port>.<intf-id>
CLI(config-if)> dialer watch-group <list-name>
```

The connect timer permits the backup interface to be open only few seconds after the routes/interfaces in the watch-list have disappeared. This dampens the transient effect of a route/interface missing few seconds that should not cause the interface opening (for example, unstable connection or renewal of the WAN IP address). The default value is 30 seconds. To configure the connect delay, the following command is used:

```
CLI(config-if)> dialer watch-timer connect <seconds>
```

The disconnect timer permits to close the interface few seconds after that the missing routes/interfaces in the watch-list have reappeared. By default, the interface is closed right after that the missing routes/interfaces are back. To configure the disconnect delay, the following command is used:

```
CLI(config-if)> dialer watch-timer disconnect <seconds>
```

Warning: the dialer-group must be disabled if you do not want the inactivity timer to shut down the interface periodically (command: `dialer-group idle-timeout 0` or `no dialer-group idle-timeout`).

Statistics

To display the dialer watch-group configuration, use:

```
CLI> show dialer-watch-group config
```

To display the dialer watch-group statistics:

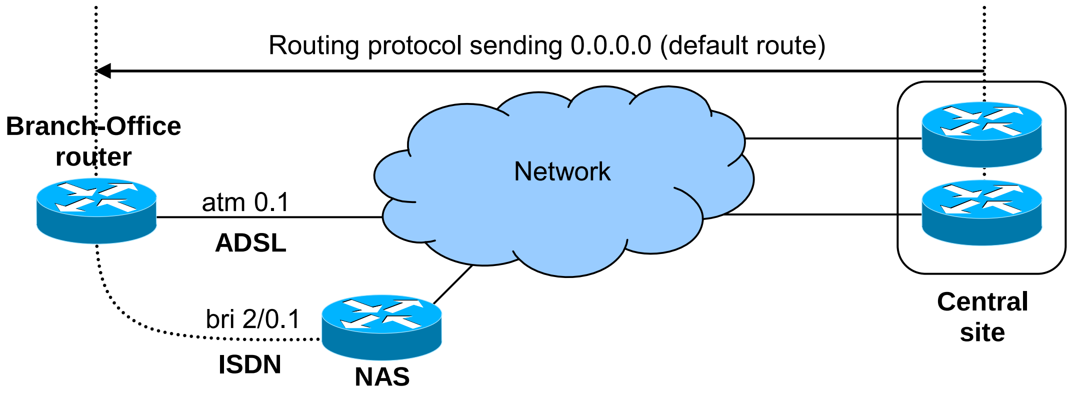
```
CLI> show dialer-watch-group statistics
```

To enable dialer watch-group traces:

```
CLI> debug dialer-watch-group
```

Example

On the following diagram, the branch-office router receives the default route on its ADSL interface (`atm 0.1`). When the router looses the connection to the central site, the default route will disappear. The default route is monitored to activate the ISDN backup. We assume that the ADSL/ATM interface is already configured as well as the BRI interface (`outgoing call` or `both-way call`).



Let us assume RIP is used to distribute the default route. The WAN interfaces are on the network 20.0.0.0/8. The following configuration commands must be added to allow the use of the dialer watch-group:

```
router rip
 passive-interface fastethernet 0/0
 network 20.0.0.0
exit

dialer watch-list default-route-monitor
 ip 0.0.0.0 0.0.0.0
exit

interface isdn-dialer 2/0.1
 dialer watch-group default-route-monitor
 dialer watch-timer disconnect 60
 no dialer-group idle-timeout
 ! Add 1 to metrics learnt via the BRI interface
 ! So that the default route on the ATM interface
 ! overwrites the default route on the BRI.
 ip rip metric 1
exit
```

14 INTERWORKING FUNCTIONS FOR LEGACY PROTOCOLS

14.1 FR - ATM INTERWORKING

This feature offers interworking functions between Frame Relay and ATM networks.

Two types of Frame Relay-ATM interworking are defined by the Frame Relay Forum and implemented in OneOS:

- FRF.5: transport of Frame Relay frames over ATM
- FRF.8: translation between Frame Relay and ATM

14.1.1 FRF.5 Frame Relay-ATM Network Interworking

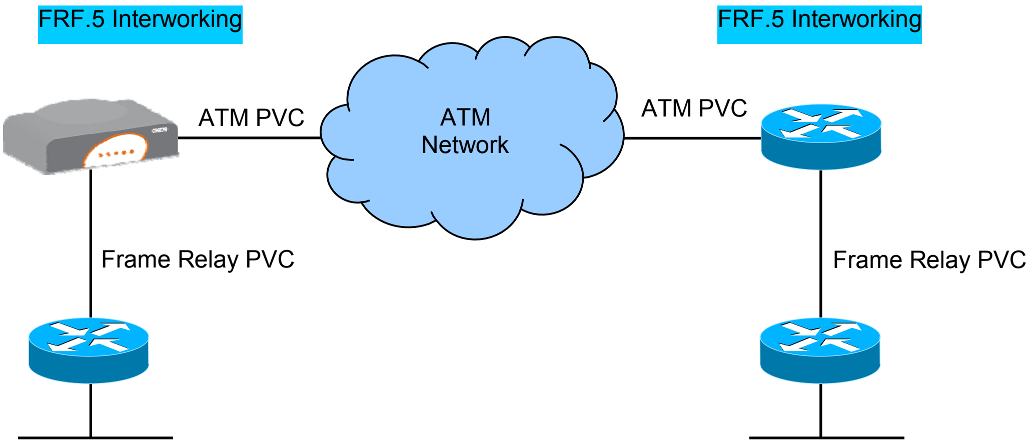
14.1.1.1 Principle

FRF.5 permits Frame Relay (FR) end-users to communicate over an intermediate ATM network. The gateway between FR and ATM supports FRF.5. As the full FR frame is transported over the ATM network, this interworking is transparent to the FR payload and functions for any FR multiprotocol extension.

When the OneOS-based router detects a fault on an ATM PVC (the intrusive ATM OAM must be enabled), its software enters into one of the following states: TP-state, VC-AIS-state, VC-RDI-state, VP-AIS-state, VP-RDI-state, VC-LOC-state, or failing loopback. When the PVC is in one of the previously mentioned state, the HDLC interface is placed out of service.

When a defect is detected on the HDLC interface, a VC-AIS is sent, thus signaling the unavailability of the CPE. The interworking between faults signaled on the HDLC interface and the ATM VC makes it possible to carry out end-to-end surveillance of the data path.

Example:



14.1.1.2 FRF.5 Configuration

Two modes are offered:

- One-to-one connection: every DLCI must be configured and the user can define specific behavior for each DLCI.
- Many-to-one connection: DLCI, which should have the same characteristics are grouped in a vc-group. Then, the connection is done for the group. The configuration steps are thus reduced for large number of DLCI.

14.1.1.3 One to one Interworking Connection

Step 1: selection of the physical frame Relay interface:

```
CLI> configure terminal
CLI(configure)> interface serial <slot>/<port>
```

Step 2: configuration of the encapsulation type:

```
CLI(config-if)> encapsulation frf
```

Step 3: selection of a switched frame relay DLCI:

```
CLI(config-if)> frame relay interface-dlci <dlci-number> switched
```

The entered parameters are validated using the following command:

```
CLI(config-if)> execute
```

14.1.1.3.1 Configuration of the ATM interface

Step 4: selection of the ATM interface:

```
CLI(configure)> interface atm <interface>.<subinterface>
```

Step 5: selection of the ATM PVC with FRF.5 encapsulation:

```
CLI(config-if)> pvc frame-relay vpi <vpi-number> vci <vci-number>
```

The entered parameters are validated using the following command:

```
CLI(config-if)> execute
```

14.1.1.3.2 Connection Frame Relay to ATM

Step 6: creation of a connection between the Frame Relay DLCI and the ATM PVC:

```
CLI(configure)> connect network-interworking <cnx-name>
                         serial <port>/<slot> <dlci-number>
                         atm <interface> vpi <vpi-number> vci <vci-number>
                         [<dlci-atm>]
```

`cnx-name` is an arbitrary string designating the connection name. `dlci-atm` is the DLCI number of the Frame Relay packet once interworked through the FRF.5 function. This parameter is optional and enables DLCI translation. If not provided, the translated DLCI value is by default the same as the one on the serial interface.

(Optional) Step 7: configuration of frame-based tagging. The ATM Cell Loss Priority (CLP) field in the ATM cell header and Discard Eligible (DE) bit mapping are configured for the configuration described in the FRF.5 standard, which also comprises EFCI to FECN mapping. CLP marking is selected the next command (Default: 0):

```
CLI(config-FRF.5)> clp-bit { 0 | 1 | map-de }
```

DE marking is enabled using the next command (Default: "`no de-bit map-clp`"):

```
CLI(config-FRF.5)> de-bit map-clp
```

Then, the parameters must be validated:

```
CLI(config-FRF.5)> execute
```

14.1.1.4 Many to one Interworking Connection

Step 1: selection of the physical frame Relay interface:

```
CLI> configure terminal
CLI(configure)> interface serial <slot/port>
```

Step 2: selection of the encapsulation type:

```
CLI(config-if)> encapsulation frf
```

Step 3: declaration of the switched frame relay DLCI:

```
CLI(config-if)> frame relay interface-dlci <dlci-number> switched
```

Validation of the parameters:

```
CLI(config-if)> execute
```

Step 4: Creation and configuration of a VC group.

The `vc-group` is the entity gathering DLCI, which will be processed similarly (same ATM PVC, same frame-based tagging). First, the `vc-group` is declared:

```
CLI(configure)> vc-group <group name>
```

A DLCI is added to this group using the following command:

```
CLI(config-vc-group)> serial <slot/port> <dlci-serial> [<dlci-atm>]
```

`dlci-serial` is the selected DLCI on the serial port. The DLCI can be translated on the ATM side in `dlci-atm`. This last command argument is optional; if not specified, the default value is taken.

14.1.1.4.1 Configuration of the ATM interface

Step 5: selection of the ATM interface:

```
CLI(configure)> interface atm <interface>.<subinterface>
```

Step 6: selection of the ATM PVC with FRF.5 encapsulation:

```
CLI(config-if)> pvc frame-relay vpi <vpi-number> vci <vci-number>
CLI(config-if)> execute
```

14.1.1.4.2 Connection Frame Relay to ATM

Step 7: selection of a connection between a Frame Relay DLCI group and an ATM PVC

```
CLI(configure)> connect vc-group <cnx-name> <group-name>
                           atm <interface> vpi <vpi-number> vci <vci-number>
```

Step 8 (optional): configuration of frame-based tagging. The ATM Cell Loss Priority (CLP) field in the ATM cell header and Discard Eligible (DE) bit mapping are configured for the configuration described in the FRF.5 standard, which also comprises EFCI to FECN mapping. CLP marking is selected the next command (Default: 0):

```
CLI(config-FRF.5)> clp-bit { 0 | 1 | map-de }
```

DE marking is enabled using the next command (Default: no `de-bit map-clp`):

```
CLI(config-FRF.5)> de-bit map-clp
```

The parameters must then be validated:

```
CLI(config-FRF.5)> execute
```

14.1.1.5 Configuration example

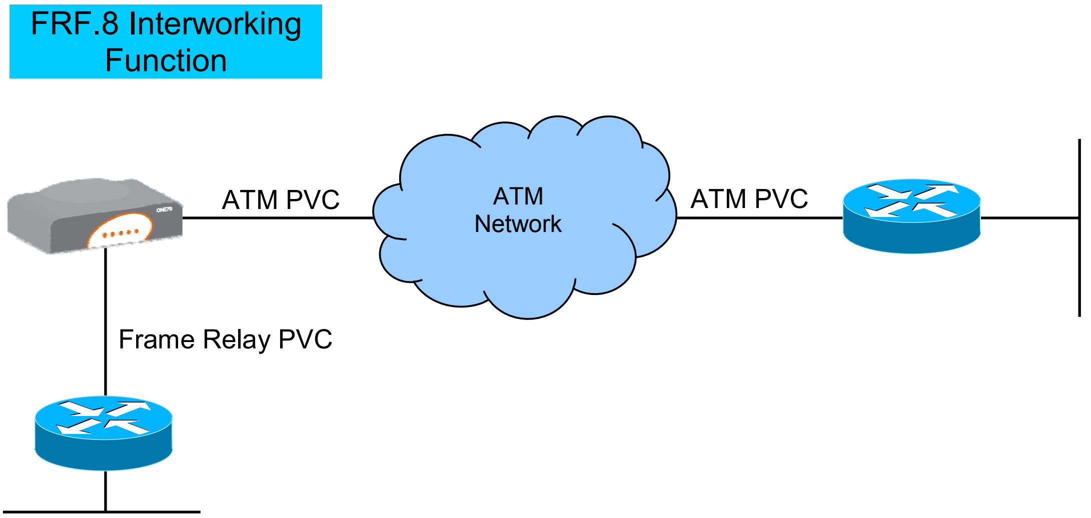
! Device interface configuration

```
hostname ONE50
interface FastEthernet 0/0
ip address 200.16.1.20 255.255.255.0
exit
interface atm 0
driver ident 0
max channels 5
max vc 5
range vp min 0 max 0
range vc min 32 max 36
pvc management up-down
execute
exit
gshdsl
execute
exit
exit

! ATM interface configuration
interface atm 0.1
pvc frame-relay vpi 0 vci 35
qos cbr pcr 2000000
priority 2
oam-pvc manage
oam end-loopback
execute
exit
exit

! Serial interface configuration
interface serial 0/0
physical-layer
no verify
exit

! FR configuration on serial interface
encapsulation frf
frame-relay interface-dlci 100 switched
frame-relay interface-dlci 101 switched
execute
exit

! VC group configuration
vc-group MyGroup
serial 0.0 100 100
serial 0.0 101 101
exit
connect vc-group MyConnexion MyGroup atm 0 vpi 0 vci 35
execute
exit
```

14.1.2 FRF.8 Interworking

14.1.2.1 Principle

This service specified by the Frame Relay Forum allows a Frame Relay end-user to communicate with an ATM end-user.



14.1.2.2 FRF.8 Configuration

The LMI on the HDLC interface is managed by default, in order to signal unavailability of a single VC to the corresponding DLCI while other VCs remain operational. The surveillance of the ATM VC state is activated when the "`intrusive ATM OAM`" is activated. The LMI NUI is managed. The LMI version is compliant with ITU-T Q.933 Annex A.

Because the LMI is configured as a NUI, asynchronous status reports are not sent spontaneously. The device will reply to full status report requests received (on DLCI #0) and to requests for Link Integrity Verification (LIV).

14.1.2.3 Serial Interface Configuration

Step 1: selection of the frame Relay interface:

```
CLI> configure terminal
CLI(configure)> interface serial <slot>/<port>
```

Step 2: selection of the encapsulation type:

```
CLI(config-if)> encapsulation frf
```

Step 3: selection of a switched frame relay DLCI:

```
CLI(config-if)> frame relay interface-dlci <dlci-number> switched
CLI(config-if)> execute
```

14.1.2.4 Configuration of the ATM interface

Step 4: selection of the ATM interface:

```
CLI(configure)> interface atm <interface>.<subinterface>
```

Step 5: selection of the ATM PVC supporting the FRF.8 interworking function:

```
CLI(config-if)> pvc fr-atm-srv vpi <vpi-number> vci <vci-number>
CLI(config-if)> execute
```

14.1.2.5 Connection Frame Relay to ATM

Step 6: creation of a connection between the Frame Relay DLCI and the ATM PVC

```
CLI(configure)> connect service-interworking <cnx-name>
                serial <slot>/<port> <dlci-number>
                atm <interface> vpi <vpi-number> vci <vci-number>
```

FRF.8 actually provides two operating modes. The translation mode removes the FR header and translates the encapsulation over ATM. The transparent mode keeps part of the Frame Relay header, which ensures no protocol-specific information is lost after encapsulation over ATM. This mode is useful for protocols where no standard ATM encapsulation exists (such as VoFR or X.25). By default, the FRF.8 is set in translation mode. If the FRF.8 transparent mode must be implemented, use the following command:

```
CLI(configure)> no service translation
```

Use `service translation` to re-activate the translated FRF.8 mode.

Step 7 (optional): configuration of frame-based tagging. The ATM Cell Loss Priority (CLP) field in the ATM cell header and Discard Eligible (DE) bit mapping are configured for the configuration described in the FRF.8 standard. CLP marking is selected the next command (Default: 0):

```
CLI(config-FRF.8)> clp-bit { 0 | 1 | map-de }
```

DE marking is enabled using the next command (Default: `no de-bit map-clp`):

```
CLI(config-FRF.8)> de-bit { 0 | 1 | map-clp }
```

EFCI marking is enabled as follows:

```
CLI(config-FRF.8)> efci-bit { 0 | 1 | map-fecn }
```

Then, the parameters must be validated:

```
CLI(config-FRF.8)> execute
```

14.1.2.6 Configuration example

```
configure terminal
hostname ONE50
interface FastEthernet 0/0
ip address 200.16.1.20 255.255.255.0
exit
interface atm 0
driver ident 0
max channels 3
max vc 3
range vp min 0 max 0
range vc min 32 max 34
execute
exit
gshdsl
execute
exit
exit
interface atm 0.1
pvc fr-atm-srv vpi 0 vci 32
qos cbr pcr 1000000
priority 2
execute
exit
exit
interface atm 0.2
pvc fr-atm-srv vpi 0 vci 34
qos cbr pcr 1000000
priority 2
execute
exit
exit
interface atm 0.3
pvc pppoa vpi 0 vci 33
ip address 20.16.1.20
ipcp static
authentication no
qos cbr pcr 304000
```

```
priority 2
execute
exit
exit
interface serial 0/0
physical-layer
no verify
exit
encapsulation frf
frame-relay interface-dlci 16 switched
frame-relay interface-dlci 17 switched
execute
exit
connect service-interworking FRF.8_1 serial 0/0 16 atm 0 vpi 0 vci 32
execute
exit
connect service-interworking FRF.8_2 serial 0/0 17 atm 0 vpi 0 vci 34
execute
exit
ip route 220.16.1.0 255.255.255.0 20.16.1.3
exit
```

14.1.2.7 Status and Statistics Commands

To display the serial link configuration use the following command:

```
CLI> show configuration interface serial <slot>/<port>
```

Example:

```
CLI> show configuration interface serial 0/0
```

Serial interface slot 0 port 0

```
Physical layer mode: Autodetect
Physical modulation type: Autodetect
105 signal: default
106 signal: default
107 signal: default
108 signal: default
109 signal: default
Rate: 2000000
MTU: 1500
Physical layer sync: Sync
Physical layer sync mode: nrz
Activity link timer: 1
Activity link count: 3
Encapsulation: FRF
Frame relay interface dlci: 100
Frame relay interface dlci: 101
Interface status: DOWN
Configuration status: Completed
```

To display statistics regarding the interworking service use the following command:

```
CLI> show statistics frfiwf
```

Example:

```
CLI> show statistics frfiwf
```

```
+-----------------------------------------------------------------+
| Statistics for FRFIWF                                                        |
+-----------------------------------------------------------------+
+-----------------------------------------------------------------+
| SERIAL statistics                                                            |
+-----------------------------------------------------------------+
| 3 DLCI registered                                                            |
| 65229 frames received                                                        |
| 0 frames rejected                                                            |
+-----------------------------------------------------------------+
| DLCI 0 (Connected to DLCI 0 on Serial)                                               |
| Connexion state is UP                                                                |
| Bit DE is mapped to CLP                                                              |
| Bit FECN is mapped to DLCI                                                           |
| 12585 frames bridged                                                                 |
| 24 frames dropped                                                                    |
+-------------------------------------------------------------------+
| DLCI 100 (Connected to DLCI 100 on Serial)                                           |
| Connexion state is UP                                                                |
| Bit DE is mapped to CLP                                                              |
| Bit FECN is mapped to DLCI                                                           |
| 24861 frames bridged                                                                 |
| 1941 frames dropped                                                                  |
+-------------------------------------------------------------------+
| DLCI 101 (Connected to DLCI 101 on Serial)                                           |
| Connexion state is UP                                                                |
| Bit DE is mapped to CLP                                                              |
| Bit FECN is mapped to DLCI                                                           |
| 23895 frames bridged                                                                 |
| 1925 frames dropped                                                                  |
+-------------------------------------------------------------------+
| ATM 0/ 35 statistics                                                                 |
+-------------------------------------------------------------------+
| 3 DLCI registered                                                                    |
| 32610 frames received                                                                |
| 0 frames rejected                                                                    |
+-------------------------------------------------------------------+
| DLCI 0 (LMI server)                                                                  |
| Connexion state is UP                                                                |
| Bit DE is mapped to CLP                                                              |
| Bit FECN is mapped to DLCI                                                           |
| 18659 frames bridged                                                                 |
| 3 frames dropped                                                                     |
+-------------------------------------------------------------------+
| DLCI 100 (Connected to DLCI 100 on Serial)                                           |
| Connexion state is UP                                                                |
| Bit DE is mapped to CLP                                                              |
| Bit FECN is mapped to DLCI                                                           |
| 7133 frames bridged                                                                  |
| No errors                                                                            |
+-------------------------------------------------------------------+
| DLCI 101 (Connected to DLCI 101 on Serial)                                           |
| Connexion state is UP                                                                |
| Bit DE is mapped to CLP                                                              |
| Bit FECN is mapped to DLCI                                                           |
| 6818 frames bridged                                                                  |
| No errors                                                                            |
+-------------------------------------------------------------------+
RX FIFO informations

AAL5 RX FIFO:
Nb current element: 0/256 (0%)
Nb max element: 10/256 (3%)
Nb time fifo was full: 0
Nb loop: 77067
Nb threshold reached: 0

FR RX FIFO:
Nb current element: 0/200 (0%)
Nb max element: 9/200 (4%)
Nb time fifo was full: 0
Nb loop: 65119
Nb threshold reached: 0
```

To display statistics of an FRF.8 connection, use the following command:

```
CLI> show statistics pvc fr-atm-srv <atm-interface-number>
                     vpi <vpi-number> vci <vci-number> dlci <dlci-number>
```

To display statistics of an FRF.5 connection, use the following command:

```
CLI> show statistics pvc frame-relay <atm-interface-number>
                     vpi <vpi number> vci <vci number> dlci <dlci-number>
```

Example:

```
CLI> show statistics pvc frame-relay 0 vpi 0 vci 35 dlci 100
+-------------------------------------------------------------+
| Statistics for DLCI 100 on VP/VC 0/35                                         |
+-------------------------------------------------------------+
| Line status is UP                                                             |
| PVC on network side is UP                                                     |
| Connexion state is UP                                                                |
| 0 frames received                                                                    |
| 0 frames bridged                                                                     |
+--------------------------------------------------------------+
| No errors                                                                            |
+--------------------------------------------------------------+
```

To display statistics regarding the FR-ATM interworking connections use this command:

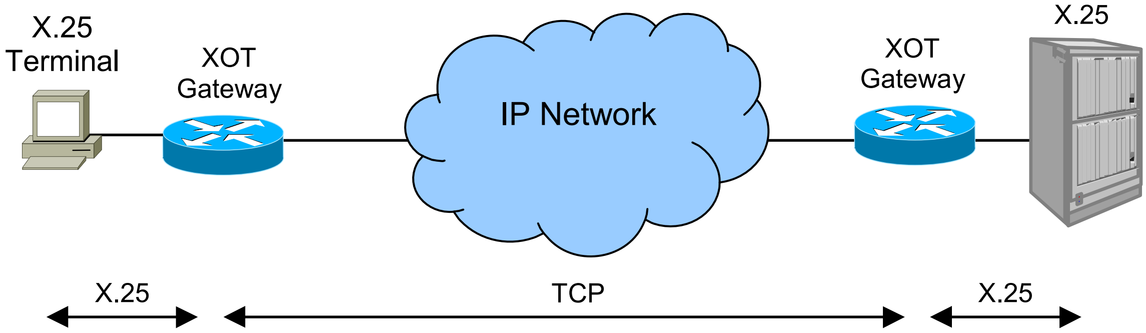
```
CLI> show connection
```

Example:

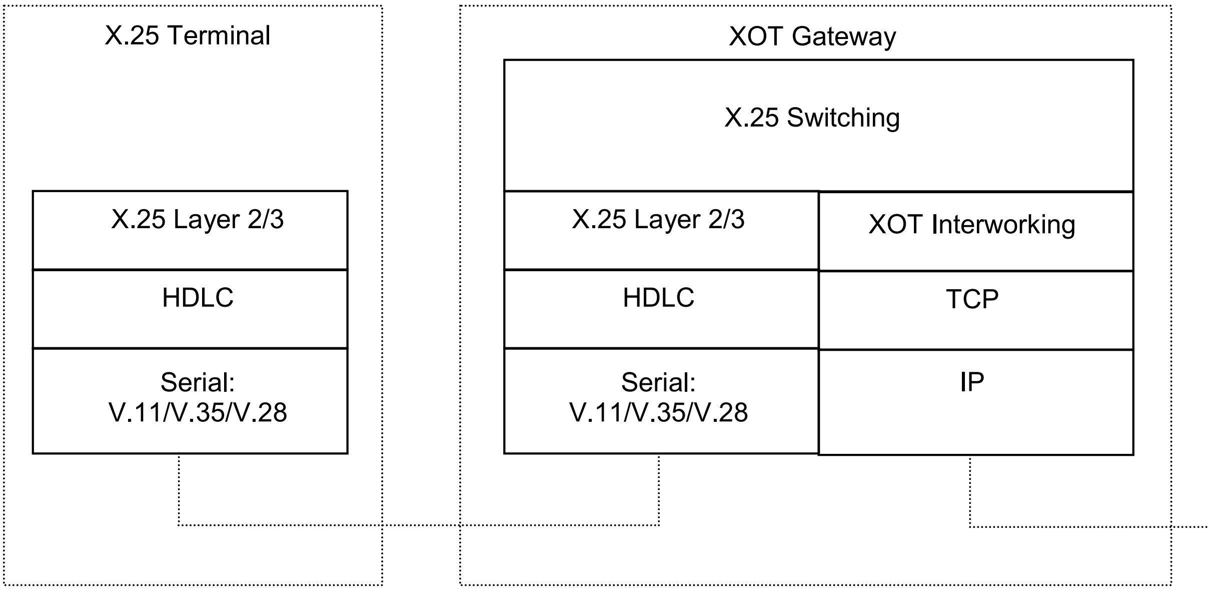
```
CLI> show connection

ID Name Segment 1 Segment 2 State
=================================================
1 FRF.5_1 Serial 0/0 100 ATM 0.1 0/35 100 UP
2 FRF.5_2 Serial 0/0 101 ATM 0.1 0/35 101 UP
```

14.2 X.25 OVER TCP/IP (XOT)

IP networks enable the use of a single infrastructure type enabling the convergence of multiple protocols and applications. The purpose of XOT (RFC 1613) is to transport X.25 packets over IP networks. The X.25 backbone network is replaced by IP. The XOT standard defines an encapsulation of X.25 packets as TCP payload. Remote endpoints connected to a XOT gateway still use X.25 natively. OneOS provides an XOT Gateway function.



The protocol stack is organized as follows:



X.25 is a connection-oriented protocol, which is similar in some extent to TCP in the IP world. When a X.25 virtual circuit must be established using X.25 signaling, a corresponding TCP session (TCP port: 1998) is created. X.25 packets are transported in that TCP session. The TCP session is then created when receiving a X.25 call setup. When the TCP session is open, the X.25 virtual circuit can be established. The TCP session must remain operational as long as the X.25 circuit is not cleared. There must be a consistency between TCP and X.25 virtual circuit state:

- A TCP disconnection results in disconnecting the X.25 circuit
- X.25 call clearing results in TCP disconnection

Layer-3 X.25 packets are sent transparently inside the XOT header. However, the OneOS XOT Gateway service manages locally X.25 layer-2/3 flow control. Indeed, TCP already has its own flow control and relaying X.25 flow control would result in unnecessary traffic and poor performance.

X.25 circuits are setup dynamically by calling X.121 addresses designating the remote X.25 endpoints. Therefore, a routing table is required to convert X.121 addresses into remote XOT gateway IP addresses.

If the call setup is not achieved properly, the routing table allows the definition of backup addresses and route calls to the alternate XOT gateways if the main XOT gateway does not respond.

Instead of using an internal routing table, OneOS is also able to query a DNS server which provides an X.121 address ←→ IP address translation function.

14.2.1 Configuration Commands

The XOT gateway configuration requires several configuration steps:

- Serial interface configuration
- X.25 Layer-2 configuration
- X.25 Layer-3 configuration
- X.121 <---> IP routing table

14.2.1.1 Serial Interface

In global configuration mode, the serial interface shall be declared:

```
CLI(configure)> interface serial <slot>/<port>
```

The serial interface can be shutdown:

```
CLI(config-if)> [no] shutdown
```

If some parameters must be set for the physical interface, enter then:

```
CLI(config-if)> physical-layer
```

Refer to section 4.1 for details about physical layer configuration. In serial interface configuration mode, the following command declares the serial port in X.25 mode:

```
CLI(config-if)> encapsulation x25
CLI(x25)>
```

14.2.1.2 X.25 Layer-2

To enter layer-2 configuration parameters, use the following command:

```
CLI(x25)> lapb { dte | dce }
CLI(x25 lapb)>
```

The default value is `dce`. In this configuration mode, several parameters can be adjusted:

```
CLI(x25 lapb)> modulo { 8 [K <1..7>] | 128 [K <1..127>] }
```

Configures the numbering size: it can be numbered modulo 8 (default). The K window can be adjusted (should be set high for long transit delays). When the numbering is modulo 8, K is by default 7, otherwise it is 30.

```
CLI(x25 lapb)> N2 <1..255>
```

Provides the maximum number of retries for a frame unacknowledged by the remote side; Default: 10 retries.

```
CLI(x25 lapb)> T1 <1..640>
```

Provides in tenths of seconds the retransmission timer indicating the maximum delay after which the LAPB protocol will retransmit an unacknowledged frame. Default: 30 sec.

```
CLI(x25 lapb)> T2 <1..160>
```

Provides in tenths of second (default is 7) the maximum delay after which the LAPB will acknowledge a received frame. T2 is usually T1/4 frame acknowledge timer.

When parameters are set, enter the `exit` command.

14.2.1.3 X.25 Layer-3 Configuration

To enter in X.25 packet configuration mode, enter the following command, beginning in X.25 interface configuration mode:

```
CLI(x25)> packet { dte | dce }
CLI(x25 packet)>
```

The default behavior is the `dce` mode. Then, in this configuration mode you can enter some additional layer-3 parameters:

```
CLI(x25 packet)> lic <0..4095>
```

- (Default is 0)
- We can define the range of values for the identifier designating the X.25 logical channel. The above command sets the lowest incoming circuit number.

```
CLI(x25 packet)> hic <0..4095>
```

- (Default is 0)
- Highest incoming circuit number (0 means no check on the upper value).
- `hic` must be equal or greater than `lic`.

```
CLI(x25 packet)> ltc <0..4095>
```

- (Default is 1)
- Lowest two-way circuit number.

```
CLI(x25 packet)> htc <0..4095>
```

- (Default is 8)
- Highest two way circuit number (0 means no check on the upper value).
- `htc` must be equal or greater than `ltc`.

```
CLI(x25 packet)> loc <0..4095>
```

- (Default is 0)
- Lowest outgoing circuit number (0 means no such circuit).

```
CLI(x25 packet)> hoc <0..4095>
```

- (Default is 0)
- Highest outgoing circuit number (0 means no such circuit)
- hoc must be equal or greater than `loc`.

The maximum number of Virtual Circuit is checked when executing the `execute` CLI command. When entering the `execute` command, some memory space is reserved to guaranty X.25 call establishment. If the maximum is reached (50 VC) the configuration is refused with an error message.

```
CLI(x25 packet)> [no]cug <00..99>
```

- Selects a Closed User Group (CUG) to which the X25 subscriber will be part of.
- A subscriber may belong to up to 30 CUG.

```
CLI(x25 packet)> [no]x25-address <x121-address>
```

- X.121 address of the X.25 subscriber (serial interface). This parameter can be used to route incoming XOT calls to the proper X.25 serial interface.

- When present, this address must be inserted as calling-address in incoming call packets if initially not present in the call packet received from the X.25 subscriber.

```
CLI(x25 packet)> [no] x25-outgoing suppress called-address
```

- Selects or not the option of called address suppression in the call packet format (sent on the X.25 interface).
- When present, the called address must be suppressed in the outgoing call packet. Default: disabled.
- This command has a higher priority than any conversion command (because it is applied after any conversion).

```
CLI(x25 packet)> [no] x25-outgoing suppress calling-address
```

- Selects or not the option of calling address suppression in the call packet format (sent on the X.25 interface). When present, the calling address must be suppressed in the outgoing call packet.

```
CLI(x25 packet)> [no] x25-outgoing convert called-address <match-mask>
                                                          [<string>]
```

This command converts the X.121 called address field for X25 calls issued to terminal for X25 calls issued to terminal (outgoing calls from OneOS X25 interface). This command has a lower priority than any suppression commands (which are applied before). Only one conversion can be defined per interface.

The first parameter `<match-mask>` is mandatory and allows selecting the addresses on which conversion is applicable. This parameter is a string with the following characters:

- any digits from 0 to 9: this characters must match exactly
- '#': any (non-void) digit matches
- '*': it can be entered only once at the end of the string. Any remaining digits or none are matched.

Examples:

<match-mask>	<x121 address>	Result
8000##	80004	no match
8000##	800044	match
8000##	8000444	no match
8000##*	8000444	match
8#4#*	804599999	match

The second parameter `<string>` is optional and is the replacement string. If not specified, the command permits to selectively suppress the whole called address. This parameter is a string with digits from 0 to 9, and/or with '#' character(s) and/or with one '*' character. If present, the '*' character must be placed at the end.

The following rules are applied for conversion:

- the X121 address to convert is explored digit by digit
- a '*' character must be seen as all remaining characters are '*'
- at the same rank in `match-mask` and in `string`, the different combinations of characters give the conversion to apply on this digit:

Called Nbr	<match-mask>	<string>	Conversion
tgt_digit	tgt_digit / #	repl_digit	=> repl_digit
tgt_digit	tgt_digit / #	#	=> tgt_digit
tgt_digit	tgt_digit / #	*	=> <<removed>>
tgt_digit	tgt_digit / #	<none>	=> <<removed>>
tgt_digit	*	repl_digit	=> repl_digit <<added>>
tgt_digit	*	#	=> tgt_digit
tgt_digit	*	*	=> tgt_digit
tgt_digit	*	<none>	=> <<removed>>
<none>	*	repl_digit	=> repl_digit <<added>>
<none>	*	<not digit>	=> <<none ->end >>

Examples:

[1] Replacing a prefix

```
x25-outgoing convert called-address ####* 8000*
```

- x121: 9000781234 => 8000781234
- x121: 60004554 => 80004554

[2] Removing a prefix

```
x25-outgoing convert called-address ######* *
```

- x121: 9000781234 => 1234
- x121: 60004554 => 54

[3] Insertion of prefix

```
x25-outgoing convert called-address * 8000*
```

- x121: 781234 => 8000781234
- x121: 4554 => 80004554
- x121: 54 => 800054
- x121: (no address) => 8000

[4] Insertion of full address

```
x25-outgoing convert called-address * 8000541234
```

- x121: 9000781234 => 8000541234
- x121: 60004554 => 8000541234
- x121: 54 => 8000541234
- x121: (no address) => 8000541234

[5] Address suppression

```
x25-outgoing convert called-address *
```

- x121: 9000781234 => (no address)
- x121: 60004554 => (no address)
- x121: (no address) => (no address)

[6] Replacing intermediate digits by others with insertion

```
x25-outgoing convert called-address 9###64* 8###8899*
```

- x121: 9000641234 => 800088991234
- x121: 900064 => 80008899

[7] Adding digits at the end (fixed length addresses)

```
x25-outgoing convert called-address ########* ########99
```

- x121: 90006442 => 9000644299
- x121: 70004401 => 7000440199

[8] Removing digits at the end

```
x25-outgoing convert called-address ########* ########
```

- o x121: 900064421234 => 90006442
- o x121: 7000440188 => 70004401

```
x25-outgoing convert called-address ######## ######
```

- o x121: 90001234 => 900012
- o x121: 70004488 => 700044

```
CLI(x25 packet)> [no] x25-incoming convert calling-address  <match-mask>
                                                            [<string>]
```

This command converts the X.121 calling address field for X.25 calls issued from a terminal (incoming calls to OneOS X.25 interface). This command is applied before any local X.121 address insertion i.e. if the result of conversion is “no calling address”, and if the local address has been configured, it will be added. This command can only be defined once per interface.

The first parameter `match-mask` is mandatory and allows selecting the addresses on which conversion can apply and, for these addresses the part of the address which will be converted (in combination with `string` parameter). See detailed comment in "`x25-outgoing convert called-address`"

The second parameter `string` is optional and is the replacement string. If it is not specified this is equivalent to suppress the full x121 address. This parameter is a string with digits from 0 to 9, and/or with '#' characters, and/or with one '*' character. If present the '*' character must be at the end. See detailed comment in "`x25-outgoing convert called-address`".

```
CLI(x25 packet)> [no] x25-incoming insert cug <cug-value>
```

This command inserts a Closed User Group (CUG) facility field for X25 calls issued from terminal (incoming calls to OneOS X25 interface). This command is applied before any check; if CUG checking is configured it must include this CUG. `cug-value` ranges from 0 to 99.

For the X25 packet data flow, each direction input and output may independently specify its own packet size and window size.

```
CLI(x25 packet)> ips { 16 | 32 | 64 | 128 | 256 | 512 | 1024 }
```

- o Maximum input packet size in bytes (default is 128).

```
CLI(x25 packet)> ops { 16 | 32 | 64 | 128 | 256 | 512 | 1024 }
```

- o Maximum output packet size in bytes (default is 128).

```
CLI(x25 packet)> win <1..7>
```

- o Window size for input packets (Default: 2).

```
CLI(x25 packet)> wout <1..7>
```

- o Window size for output packets (Default: 2).

The logical channel number 0 can be used or not as incoming, outgoing or two-way channel.

```
CLI(x25 packet)> enable { lcn0-as-incoming | lcn0-as-outgoing
                                          | lcn0-as-two-way }
CLI(x25 packet)> disable { lcn0-as-incoming | lcn0-as-outgoing
                                          | lcn0-as-two-way }
```

Finally, complete the configuration by entering the following commands:

```
CLI(x25 packet)> execute
CLI(x25 packet)> exit
CLI(x25)> exit
CLI(config-if)> exit
CLI(configure)>
```

14.2.1.4 X.25 Routing

The X.25 routing enables to redirect X.25 calls to an output interface. There are two cases:

- An incoming XOT call is received and its must be routed to a serial interface (the on-board serial 0/0 interface of the ONE60 or on the V.28 daughter board of the ONE60/200, namely serial 2/{1|2})
- An X.25 call is received and must be redirected to XOT

A call packet cannot be redirected on the interface that it is originating from. The configuration commands follow. To enter in X.25 routing configuration, enter:

```
CLI(configure)> [no] route x25
```

The routing table allows up to 50 entries. By default, it is forced to route X.25 packets to serial 0/0 (V.11/V.35/V.28/V.36) and vice-versa.

Then select an X.121 address that you want to attach an output interface afterwards:

```
CLI(route-x25)> [no] x121 <x121-pattern>
```

The pattern entered can be a X.121 number (example: 700011) or a number with a wild card (example: 7000* means all numbers starting with 7000). Any call with a X.121 number matching this entry is routed on the output interface

```
CLI(route-x25)> [no] serial <slot>/<port>
CLI(route-x25)> [no] xot
```

Then enter `exit` and create the next x121 entry if needed. Once you have finished the X.25 routing configuration, enter `execute`.

The routing sorts the entries out from the most discriminating number (longest number without '*') to the least discriminating number ('*'). The routing is made by matching call packets with the sorted entries.

Example:

```
route x25
 x121 8000*
  xot
 exit
 x121 9000*
  serial 0/0
 exit
 x121 90000021
  serial 2/1
 exit
 x121 90000022
  serial 2/2
 exit
 execute
exit
```

14.2.1.5 XOT Routing Table

Note: as of V4.3R2E2 software release, XOT is not supported outside default VRF.

The main goal of the XOT routing table is to define an IP destination for each X.121 address (or group of addresses, see utilization of the ***** star character) that the XOT service can process when handling outgoing X.25 call packets.

The XOT routing table is organized as follows, and contains up to 50 entries.

Each entry specifies:

- X.121 target address.
- Associated IP target addresses or DNS parameters.
- Keep alive parameters specific to the TCP sessions with the X.121 target.
- Selection method of the possible remote XOT gateway addresses.

To enter in XOT routing table configuration mode:

```
CLI(configure)> route xot
CLI(route-xot)>
```

If you do not define the source address of the XOT gateway, the source address is the same as the outgoing interface address. This can be inconvenient, as the interface address is not always known in advance (for example, the IP address of a PPP interface can be retrieved dynamically). You can then choose a fixed IP address as follows:

```
CLI(route-xot)> ip address <source-ip-addr>
```

Alternatively, you can select the IP address of another interface (which IP address is usually fixed):

```
CLI(route-xot)> ip unnumbered <interface-type> <unit>
```

Then for each entry of the table associated with an X.121 address (or group of addresses), enter:

```
CLI(route-xot)> x121 <x121-address>
CLI(x121)>
```

The Remote Address is in X.121 format. For this called address, we can associate the target IP address and other destination-specific parameters.

```
CLI(x121)> ip <ip-address> [ip <ip-address2> [ip <ip-address3>
                                              [ip <ip-address4>]]]
```

First IP address is the IP address of the destination XOT peer while other IP addresses are the IP address of optional second, third and fourth possible XOT peer.

Instead of using static IP addresses, we can use domain name resolution: DNS servers are defined globally for the router (command: `ip name-server ...`) and contains a list of IP addresses for certain domain names. To enable DNS-based XOT routing, the DNS servers must be configure to return the IP addresses of XOT gateways based on domain names. Therefore, the domain names are built based on the called X.121 address. The command is the following:

```
CLI(x121)> ip dns [<string1>]#<num-d>#[<string2>]
```

For the called X.121 address matching this routing entry, a query is sent to the DNS server(s) for the domain name built with the concatenation of the following strings:

- `string1` (optional)
- First `num-d` digits of the called X.121 address. If `num-d` = 0, the whole X.121 address is taken.
- `string2` (optional)

Example:

```
CLI(route-xot)> x121 8000*
CLI(x121)> ip dns xotresolv#3#.com
```

For the called X.121 address 80003, the `xotresolv800.com` domain name is then queried.

Beware of conflicts with the host name. Example: The host is configured as follows:

```
ip host xot600011gtw2 192.168.200.101
```

The DNS server is then configured:

```
ip domain-name xotdomain.org
ip name-server 127.0.0.1
ip dns-proxy dns-server 193.1.1.109
```

If you do not wish to use a DNS cache, use:

```
no ip dns-proxy cache
```

When querying the string `xot800010gtw1`, it returns:

```
xot800010gtw1 192.168.200.101 192.168.201.101 192.168.201.102
                                              192.168.201.103
```

As you can see, the local host is queried first, thus creating potentially a conflict.

Whether the IP address is returned through DNS servers or manually configured, we can then define the selection algorithm for XOT gateway IP addresses.

```
CLI(x121)> gateway-selection { backup | random }
```

`backup` is the default value. When `backup` is used, the first configured IP address or the first address in the list returned by DNS is selected as primary XOT gateway. If no XOT connection can be done, the second one is taken and so on.

If `random` is selected, one address among the N possible addresses is randomly selected. If no XOT connection can be built with the corresponding XOT gateway, this IP address is considered invalid. (N-1) addresses remain. The same process is repeated with the (N-1) addresses, then N-2 ... until all addresses are considered invalid.

If no connection is possible with all addresses, the X.25 connection is released.

```
CLI(x121)> [no] keepalive-period <1..3600>
```

- Defines in seconds (default 60) the keepalive period of the TCP connection.
- The no form of the command disables the keepalive mechanism.

```
CLI(x121)> [no] keepalive-tries <1..10>
```

- Defines the number of consecutive keepalive messages (default 4) with no acknowledgement to consider the TCP connection as silent. When the TCP connection is silent, then the TCP connection and associated X.25 circuit are cleared.
- The no form of the command disables the retry mechanism.

To terminate the entry:

```
CLI(x121)> exit
CLI(route-xot)>
```

When all the X.121 entries are set, the routing table is built by entering the following command:

```
CLI(route-xot)> execute
```

To clear an entry of XOT routing table:

```
CLI(route-xot)> no x121 <x121-address>
```

Warning: all the data of the specified entry will be lost.

To clear the XOT routing table:

```
CLI(configure)> no route xot
```

Warning: all the data of the XOT routing table will be lost.

The rules for X121 address specification are the following:

- 15 digits maximum. Example: 123456789012345.
- N digits ended by a star character masking all the subsequent characters. Example: 1234***** means all X.121 addresses from 12340 (5 digits) to 123499999999999 (15 digits).
- Single ***** (star) character. Example: ***** means all X.121 addresses from 0 (1 digit) to 999999999999999 (15 digits).

The list of X.121 address is automatically organized by the CLI; the most discriminating address first, the least discriminating address being last. Example:

```
1234
1234*
12*
1
*
```

14.2.1.6 Statistics and Configuration Display

To display all configuration parameters of the serial interface, including default values, use the following command:

```
CLI> show configuration interface serial 0/0
Serial interface slot 0 port 0
Physical layer mode: Autodetect
Physical modulation type: Autodetect
105 signal: default
106 signal: default
107 signal: default
108 signal: default
109 signal: default
Rate: 2000000
MTU: 1500
Physical layer sync: Sync
Physical layer sync mode: nrz
Activity link timer: 1
Activity link count: 3
Encapsulation: X25
LAPB DTE modulo 8 windows(k) = 7
n2 = 10 T1 = 1600 ms T2 = 400 ms
PACKET DTE: X121 address =
0000 incoming circuits from 0000 to 0000
0025 two ways circuits from 0001 to 0025
0000 outgoing circuits from 0000 to 0000
default packet length 128 (input) 128 (output)
default packet windows 2 (input) 2 (output)
no CUG to check on incoming call
Interface status: DOWN
Configuration status: Completed
```

To display serial interface statistics, the command is:

```
CLI> show statistics serial 0/0

Statistics for LINK LEVEL LAYER (LAPB)

          [ SABM(E) ]          [ DISC ]            [ UA ]
     .start    .lastread .start    .lastread .start    .lastread
Tx : 000000004 000000000 000000000 000000000 000000000 000000000
Rx : 000000000 000000000 000000000 000000000 000000001 000000000
          [ DM ]               [ FRMR ]            [ REJ ]
     .start    .lastread .start    .lastread .start    .lastread
Tx : 000000000 000000000 000000000 000000000 000000000 000000000
Rx : 000000001 000000000 000000000 000000000 000000000 000000000
          [ DATA ]              [ RR ]             [ RNR ]
     .start    .lastread .start    .lastread .start    .lastread
Tx : 000000019 000000018 000000002 000000001 000000000 000000000
Rx : 000000017 000000016 000000005 000000004 000000000 000000000
Rx ignored frames 000000000
Statistics for PACKET LAYER

          [ CALL ]            [ CLEAR ]        [ RESTART ]
     .start    .lastread .start    .lastread .start    .lastread
Tx : 000000000 000000000 000000000 000000000 000000001 000000000
```

```
Rx : 000000001 000000001 000000001 000000001 000000001 000000000
        [ CALL CONF ]        [ CLEAR CONF ]   [ RESTART CONF ]
     .start    .lastread .start    .lastread .start    .lastread
Tx : 000000001 000000001 000000001 000000001 000000000 000000000
Rx : 000000000 000000000 000000000 000000000 000000000 000000000
          [ DATA ]            [ RESET ]         [ INTERRUPT ]
     .start    .lastread .start    .lastread .start    .lastread
Tx : 000000016 000000016 000000000 000000000 000000000 000000000
Rx : 000000000 000000000 000000000 000000000 000000000 000000000
          [ RR ]            [ RESET CONF ]      [ INTR.CONF ]
     .start    .lastread .start    .lastread .start    .lastread
Tx : 000000000 000000000 000000000 000000000 000000000 000000000
Rx : 000000014 000000014 000000000 000000000 000000000 000000000
          [ DIAG ]           [ UNKNOWN ]          [ RNR ]
     .start    .lastread .start    .lastread .start    .lastread
Rx : 000000000 000000000 000000000 000000000 000000000 000000000
```

Column `.start`: number of frames / packets since the X25 is started

Column `.lastread`: number of frames / packets since the last show statistics command

The XOT statistics are provided in the serial interface statistics.

To see the XOT statistics a `show statistics` command is available and includes:

```
CLI> show statistics xot

XOT TCP statistics start lastread
incoming TCP calls 000000001 000000001
outgoing TCP calls OK 000000000 000000000
outgoing TCP calls FAIL. 000000000 000000000
outgoing XOT packets 000000014 000000014
incoming XOT packets OK 000000015 000000015
incoming XOT bad packets 000000000 000000000

Statistics for PACKET LAYER

     [       CALL      ] [      CLEAR      ] [     RESTART     ]
     .start    .lastread .start    .lastread .start    .lastread
Tx : 000000000 000000000 000000000 000000000 000000000 000000000
Rx : 000000001 000000001 000000001 000000001 000000000 000000000
     [    CALL CONF    ] [    CLEAR CONF   ] [   RESTART CONF   ]
     .start    .lastread .start    .lastread .start    .lastread
Tx : 000000001 000000001 000000001 000000001 000000000 000000000
Rx : 000000000 000000000 000000000 000000000 000000000 000000000
     [       DATA      ] [      RESET      ] [    INTERRUPT    ]
     .start    .lastread .start    .lastread .start    .lastread
Tx : 000000015 000000015 000000000 000000000 000000000 000000000
Rx : 000000000 000000000 000000000 000000000 000000000 000000000
     [        RR       ] [    RESET CONF   ] [    INTR.CONF    ]
     .start    .lastread .start    .lastread .start    .lastread
Tx : 000000000 000000000 000000000 000000000 000000000 000000000
Rx : 000000013 000000013 000000000 000000000 000000000 000000000
     [       DIAG      ] [     UNKNOWN     ] [       RNR       ]
     .start    .lastread .start    .lastread .start    .lastread
Rx : 000000000 000000000 000000000 000000000 000000000 000000000
```

Column `.start`: number of packets since the XOT is started

Column `.lastread`: number of packets since the last `'show statistics'` command

To see the XOT active connections status, a show status command is available.

```
CLI> show xot
XOT 11 active connections
id=00 LCN=0049 [connected ] remote 10.29.140.171
initiated by remote using port 1024
id=01 LCN=0050 [connected ] remote 10.29.140.171
initiated by remote using port 1049
id=02 LCN=0048 [connected ] remote 10.29.140.171
initiated by remote using port 1025
id=03 LCN=0047 [connected ] remote 10.29.140.171
initiated by remote using port 1026
id=04 LCN=0046 [connected ] remote 10.29.140.171
initiated by remote using port 1027
id=05 LCN=0045 [connected ] remote 10.29.140.171
initiated by remote using port 1028
```

```
id=06 LCN=0044 [connected ] remote 10.29.140.171
initiated by remote using port 1029
id=07 LCN=0043 [connected ] remote 10.29.140.171
initiated by remote using port 1030
id=08 LCN=0042 [connected ] remote 10.29.140.171
initiated by remote using port 1031
id=09 LCN=0041 [connected ] remote 10.29.140.171
initiated by remote using port 1032
id=10 LCN=0001 [connected ] remote 10.29.140.171
initiated by local using port 1124
```

The routing table is printed as follows:

```
CLI> show configuration xot route
XOT route: 2 routes
x121 90000010 ip 192.1.1.110
keepalive period 200 secs tries 8
x121 90000020 ip 192.1.1.120 192.1.2.120
keepalive period 60 secs tries 4
```

14.2.1.7 Debug and Traces

The following command restarts the serial interface:

```
CLI> x25 restart interface serial { 0 | 2 }/<0-2>[.<1-255>]
```

The following commands define the X.121 address and the user data (password) that will make the traffic generator generate data when called at this address with this user data. The default user data used as password is "`00GG`". The data generated is a rotating pattern of the alphanumeric characters.

```
CLI(configure)> x25 traffic-generator [called-address <x121-address>]
CLI(configure)> x25 traffic-generator password <user-data-as-password>

CLI(configure)> no x25 traffic-generator
```

For X.25 layer-3 debugging, use:

```
CLI> [no] debug x25 packets all
```

For X.25 layer-2 (LAPB) debugging, use:

```
CLI> [no] debug x25 frames all
```

For XOT SOCKET debugging (Interface between IP and XOT), use:

```
CLI> [no] debug xot socket
```

For XOT routing table debugging, use:

```
CLI> [no] debug xot routing
```

For XOT protocol debugging, use:

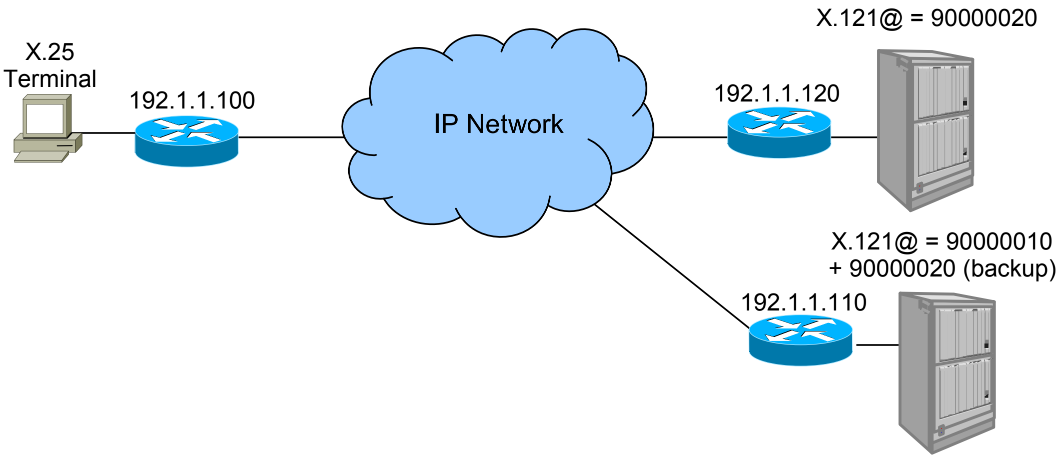
```
CLI> [no] debug xot stack
```

14.2.1.8 Configuration Example

This example provides the configuration of an OneOS-based router, which Ethernet interface (192.1.1.100) is connected to an IP network. The device is represented on the left-hand side of the following diagram. On this network, we find 2 other XOT gateways with the following addresses:

- 192.1.1.120: manages the X.121 address 90000010
- 192.1.1.110: manages the X.121 addresses 90000010 and 90000020 (backup)

Because XOT is based on TCP, the XOT flow slowly reduces its output rate when the WAN port is congested. This can be the case when the router routes UDP-based traffic from the LAN to the WAN. The use of CB-WFQ or classification of XOT in a separate CBQ class is strongly recommended. You can use the source IP of the XOT traffic as filtering criterion.



```
CLI> show running-config
Building configuration...

Current configuration:
hostname CLI
interface FastEthernet 0
ip address 192.1.1.100 255.255.255.0
exit
interface serial 0/0
encapsulation x25
lapb DTE
t1 16
n2 4
exit
packet DTE
htc 25
exit
exit
execute
route xot
x121 90000010
ip 192.1.1.110
keepalive-period 200
keepalive-tries 8
exit
x121 90000020
ip 192.1.1.120 192.1.2.110
exit
execute
exit
```

14.2.2 PAD over XOT

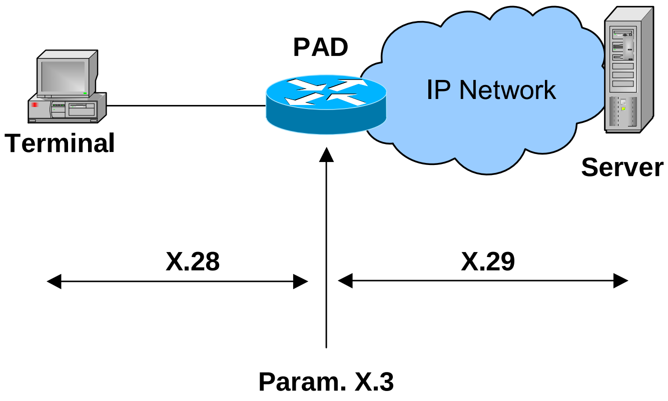
(ONE60/200 specific, only supported on the daughter-board with V.28 ports) The Packet Assembler Disassembler (PAD) function permits the communication of character-driven terminals over a packet network (generally X.25). The PAD function was implemented in OneOS to transport PAD data over X.25 over TCP/IP.

The PAD protocol is divided in three sub-components, as illustrated on the figure below:

- X.3, which defines the set of parameters describing the behavior of the router towards the terminal

- X.28, a protocol between the terminal and the PAD

- X.29, a protocol between the PAD and the remote end point.



14.2.2.1 X.3 Parameter Configuration

14.2.2.1.1 Introduction

The X.3 standard defines a set of 22 parameters. To facilitate the configuration, standardized parameter sets are pre-defined and grouped in X.3 profiles. The user can select the profile and modify few parameters if needed.

A X.3 profile can also be remotely chosen (by the terminal or remote host) using X.28 or X.29.

The parameters and their possible values are the following ('ns' means 'not supported'):

- 1: Escape character	0 none	1 DLE		
- 2: Echo	0 no echo	1 echo		
- 3: Character for packet sending		0 none	2 carriage return	126 all characters
- 4: Timer length for packet emission		0 no timer	1 to 255 n * 50 ms	
- 5: XON/XOFF Flow control (PAD to Terminal)		0 none	1 only data transfer	
- 6: Indication transmission (PAD to terminal)		0 none	1 always	
- 7: Meaning of 'break'	0 send	1 interrupt data transfer		2 reset
- 8: Send data to terminal	ns =0 always			
- 9: Padding after CR	0 never	4 four times null Char		
- 10: Force line folding	ns =0 no			
- 11: Speed	ns (read only)			
- 12: XON/XOFF flow control (Terminal to PAD)		0 none	1 always	
- 13: Insert LF after CR	ns =0 no			
- 14: Padding after LF	ns =0 none			
- 15: Character edition	ns =0 only during the command phase			
- 16: Delete character	ns =8 Backspace			
- 17: Line delete character	ns =0 no line delete			
- 18: Display line	ns =0 no display line			
- 19: Indicate edition	ns =0 no editing signal			
- 20: Echo mask	ns =0 no, all characters echoed			
- 21: Parity processing	ns =0 no parity detection and generation			
- 22: wait end of page	ns =0 no			

To ease the configuration of profiles, pre-defined values are collected in standard profiles. Here is the list of the support profiles:

Profile ID	Supported Parameters								
	1	2	3	4	5	6	7	9	12
00	1	1	126	0	1	1	2	0	0
01	0	0	0	20	0	0	2	0	0
02	0	0	0	10	1	0	21	0	0
03	1	0	2	80	0	1	21	0	0
04	1	0	2	40	0	1	21	4	0
05	1	0	2	0	1	1	21	0	0
06	1	1	126	0	1	1	21	0	0
07	1	1	2	0	1	1	0	4	0
08	1	1	2	0	1	1	21	0	0
10	1	0	2	0	0	1	21	0	0
11	0	0	0	3	0	0	21	0	0
12	1	0	126	0	1	1	2	0	0
13	1	0	0	5	1	1	21	0	0
15	1	0	0	5	0	1	21	0	0
22	1	0	0	5	1	1	21	0	1

14.2.2.1.2 Configuration

Step 1)

The X.3 parameters are collected in a virtual template. The template is based on a standard profile as shown above that you can customize. To configure the virtual template, first enter in template configuration from the global configuration terminal and select the standard profile to use:

```
CLI(configure)> [no] virtual-template x3-pad <user-profile-id>
CLI(x3-pad)> init-from <standard-profile-number>
```

The user-profile-id must be between 24 and 89.

Step 2)

If you do not need to modify the profile, go to step 3). Should you need to modify a standard profile, first enter the 'parameters' command and modify the parameters (default values are the one from the standard profile):

```
CLI(x3-pad)> parameters

CLI(x3-pad-parameters)> 01-escape-char { DLE | NONE }

CLI(x3-pad-parameters)> 02-echo { NO | YES }

CLI(x3-pad-parameters)> 03-send-char { NONE | CR | ANY }

CLI(x3-pad-parameters)> 04-send-timeout <0..255>

CLI(x3-pad-parameters)> 05-pad-flow-control { NO | YES }

CLI(x3-pad-parameters)> 06-pad-indication { No | YES }

CLI(x3-pad-parameters)> 07-break { SEND | RESET | INTERRUPT }

CLI(x3-pad-parameters)> 09-cr-padding { NO | YES }

CLI(x3-pad-parameters)> 12-terminal-flow-control { NO | YES }
```

Once all parameters whose change is required are set, enter 'exit'.

Step 3)

To complete the configuration of the template, enter `execute` and then `exit`.

14.2.2.1.3 Example

```
configure terminal
virtual-template x3-pad 50
init-from 3
parameters
04-send-timeout 50
05-pad-flow-control YES
07-break INTERRUPT
12-terminal-flow-control YES
exit
execute
exit
exit
```

14.2.2.2 V.28 Interface Configuration

First select the interface to configure (from the global configuration mode):

```
CLI(configure)> interface serial { 2/1 | 2/2 }
```

14.2.2.2.1.1 Physical Layer Configuration

Underlined parameters are default values.

If you wish to change parameters of the physical layer of the V.28 interface, enter the following commands in interface configuration mode:

```
CLI(config-if)> physical-layer
CLI(ph_layer)> async
```

The following parameters are configurable.

To set the interface speed in bit per second, use the following command:

```
CLI(async)> rate {300| 600| 1200| 2400| 4800| 9600| 38400| 57600| 115200}
```

To provide the number of data bits per byte, use the following command:

```
CLI(async)> data-bit { 5 | 6 | 7 | 8 }
```

To set the parity check type of characters, use the following command:

```
CLI(async)> parity { none | even | forced-to-0 | forced-to-1 }
```

To set the number of stop bits, use the following command:

```
CLI(async)> stop-bit { 1 | 2 }
```

To set the number of characters in the receive buffer, use the following command:

```
CLI(async)> receive-buffer-size <1..128>
```

To set the type of flow control, use the following command:

```
CLI(async)> flow-control { none | hardware | xon-xoff | both }
```

To configure the management of received "breaks", use the following command; `none` means the "break" signal is ignored, `incoming` means the received "breaks" are transferred as NULL characters in the receive buffer. The OneOS software does not generate any "break".

```
CLI(async)> break { none | incoming }
```

Enter exit 2 times once the configuration of the asynchronous level is finished:

```
CLI(async)> exit
CLI(ph_layer)> exit
CLI(config-if)>
```

14.2.2.2.2 PAD <---> X.25 Layer Configuration

To allow the serial interface to support the PAD, enter PAD as the argument for 'encapsulation':

```
CLI(config-if)> encapsulation pad
```

Then, select the PAD profile:

```
CLI(pad)> x3-profile <profile-id>
```

Where the profile-id has the following values:

- Standard X.3 profile, the support values are: from 0 to 8, from 10 to 13, 15, 22
- User-defined profiles, from 24 to 89 (it must have been created previously)

A X.121 address can be associated with the interface. It will be used as calling address in the XOT packet, when the call is initiated by the PAD:

```
CLI(pad)> [no] x121 <x121-address>
```

When the PAD must set up the XOT connection, an automatic call must be configured. If the call is not successful, the call is tried again in an infinite loop after a delay. The delay is chosen randomly between 10 and 60 seconds. The call is parameterized with a call string detailed hereafter:

```
CLI(pad)> [no] automatic-call [[[[ <facility-#1>[[[,<facility-#2>],
 ...
,<facility-#n>]-<called-x121-address>[D<user-data>]
```

`user-data` is a string with 0 to 12 characters (without '+', backspace or carriage return). As shown above, the facility list is provided separated by commas without space. The X.121 address is preceded by the '-' character without space. The facilities are written as follows:

- `N<nui>`: indicates the Network User Identifier (1 to 6 characters)
- `R`: indicates Reverse charging (the called PAD is charged, instead of the caller)
- `G<0..99>`: indicates the Closed User Group (CUG)

Example: `automatic-call Nabc123,R,G18-80000021D00GG00`

You can then define the mapping between NUI (Network User Identifier) and NUA (Network User Address). First, enter in NUI list configuration:

```
CLI(pad)> [no] nui-list
CLI(pad-nui-list)>
```

Then, configure the NUI-NUA couples as follows:

```
CLI(pad-nui-list)> [no] nui-nua <nui>.<nua-x121>
```

`nui` is a string of 16 characters maximum. Enter `exit`, when all NUI-NUA mappings are configured. Enter `exit` again when you have completed the configuration of the PAD encapsulation.

If you need to adjust XOT parameters, use the `packet dce` command. However, some default values are usually acceptable and you do not have to change these values. The default values are equivalent to the following script:

```
packet dce
lic 0
hic 0
ltc 1
```

```
htc 1
loc 0
hoc 0
no subscribe packet size
ips 128
ops 128
no subscribe window-size
win 4
wout 2
exit
```

If all parameters are configured, enter `execute`, then `exit`.

14.2.2.2.3 Statistics

To view the serial interface configuration and default values, use the following command:

```
CLI> show configuration interface serial <slot>/<port>
```

To display the interface statistics, the syntax is:

```
CLI> show statistics serial <slot>/<port>
CLI> show vxx <slot>/<port>
```

To display X.3 virtual templates:

```
CLI> show virtual-template x3-pad <template-id>
```

To display information on X.25/PAD status:

```
CLI> show x25
```

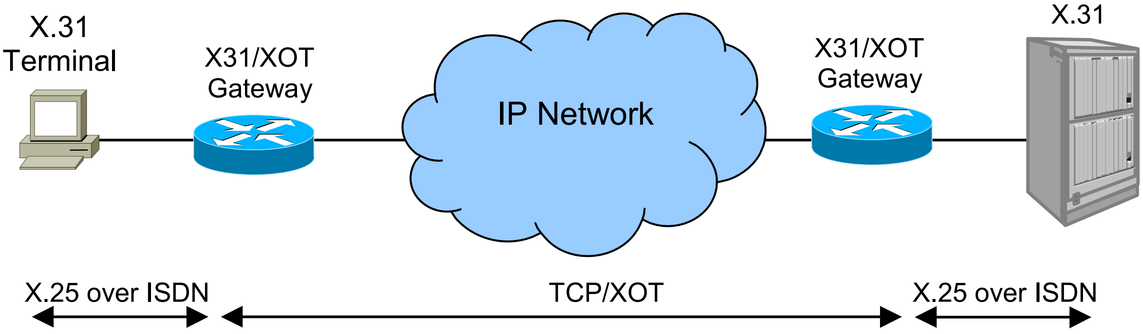
14.2.2.2.4 Example

```
configure terminal
interface serial 2/1
physical-layer
async
rate 57600
data-bit 8
parity NONE
stop-bit 1
flow-control BOTH
exit
exit
encapsulation pad
x3-profile 50
x121-address 80000021
exit
execute
no shutdown
exit
exit
```

14.3 X.25 OVER ISDN (X.31)

The purpose of X.25 over ISDN (ITU-T X.31) is to transport X.25 packets over ISDN networks. OneOS provides an X.31 Gateway function to allow X.25 communications through an ISDN link. Adding to that, using XOT (see 14.2) the X.25 backbone network is replaced by IP networks enabling the use of a single infrastructure type enabling the convergence of multiple protocols and applications.

As of V4.2R5 software release, the X.25 communication has to be established using only the D channel of an ISDN BRI interface configured in VoIP mode and there can be up to 4 PLL.



14.3.1 Configuration commands

Note that the X31 configuration is only available after having entered, using the `sw-right` command, the license key dedicated to your OneOS-based router and corresponding to the software license you bought (X31 license).

The X31 gateway configuration requires several configuration steps:

- ISDN D channel interface configuration (permanent logical link configuration)
- X.25 Layer-3 configuration (virtual template configuration)
- X.25 D channel interface configuration (X.31 configuration)
- X.121 <---> X.25 D channel interface routing

14.3.1.1 ISDN D channel interface

The permanent logical links (PLL) are declared and configured under ISDN in BRI interface mode.

In global configuration mode, enter in BRI configuration mode:

```
CLI(configure)> interface bri <slot>/<port>
```

The BRI interface can be started or shutdown:

```
CLI(conf-if)> [no] shutdown
```

Then enter in ISDN configuration mode to configure the PLL. Each PLL is defined by its static TEI number. The LAPD parameters are common to all PLL (refer to 7.2.1 for more information about LAPD parameters configuration).

```
CLI(conf-if)> isdn
CLI(isdn)> permanent-logical-link
CLI(conf-pll)> k-window <1..128>           (default = 7  )
CLI(conf-pll)> modulo-window <8|128>       (default = 128)
CLI(conf-pll)> n200-counter <1..20>          (default = 3  )
CLI(conf-pll)> t200-timer <0..63)            (default = 1  )
CLI(conf-pll)> t203-timer <0..63)            (default = 10 )
CLI(conf-pll)> [no] pll { 1 | 2 | 3 | 4 } tei <0..63>
CLI(conf-pll)> exit
CLI(isdn)> exit
CLI(conf-if)> exit
CLI(configure)>
```

14.3.1.2 X.25 layer 3 template

Templates are used to ease the configuration of X.25 layer 3 parameters. They contain a set of common parameters that can be shared across multiple X.31 interfaces.

To create and configure an X.25 layer-3 template, enter the following command in global configuration mode (refer to 14.2.1.3 for more information about X.25 layer-3 parameters configuration).

```
CLI(configure)> virtual-template x25-layer3 <id:1..32>
CLI(x25-layer-3)> packet { dte | dce }
CLI(x25-layer-3)> enable lcn0-as-incoming
CLI(x25-layer-3)> disable lcn0-as-incoming
CLI(x25-layer-3)> enable lcn0-as-outgoing
CLI(x25-layer-3)> disable lcn0-as-outgoing
CLI(x25-layer-3)> enable lcn0-as-two-way
CLI(x25-layer-3)> disable lcn0-as-two-way
CLI(x25-layer-3)> lic <0..4095>
CLI(x25-layer-3)> hic <0..4095>
CLI(x25-layer-3)> ltc <0..4095>
CLI(x25-layer-3)> htc <0..4095>
CLI(x25-layer-3)> loc <0..4095>
CLI(x25-layer-3)> hoc <0..4095>
CLI(x25-layer-3)> [no] cug <0..99>
CLI(x25-layer-3)> [no] x25-address <x121-address>
CLI(x25-layer-3)> [no] x25-outgoing convert called-address <match-mask>
                                                 [<replacement-string>]
CLI(x25-layer-3)> [no] x25-outgoing convert cug <0..99> to <0.99>
CLI(x25-layer-3)> [no] x25-outgoing insert cug <0..99>
CLI(x25-layer-3)> [no] x25-outgoing suppress called-address
CLI(x25-layer-3)> [no] x25-outgoing suppress calling-address
CLI(x25-layer-3)> [no] x25-incoming convert calling-address <match-mask>
                                                 [<replacement-string>]
CLI(x25-layer-3)> [no] x25-incoming convert cug <0..99> to <0..99>
CLI(x25-layer-3)> [no] x25-incoming insert cug <0..99>
CLI(x25-layer-3)> ips { 16 | 32 | 64 | 128 | 256 | 512 | 1024 }
CLI(x25-layer-3)> ops { 16 | 32 | 64 | 128 | 256 | 512 | 1024 }
CLI(x25-layer-3)> win <1..7>
CLI(x25-layer-3)> wout <1..7>
CLI(x25-layer-3)> [no] subscribe packet-size
CLI(x25-layer-3)> [no] subscribe window-size
CLI(x25-layer-3)> exit
CLI(configure)>
```

Note: editing an X.25 layer-3 template causes the restart of layer 3 of the X.31 interfaces using it.

To delete an X.25 layer-3 template, enter the following command in global configuration mode:

```
CLI(configure)> no virtual-template x25-layer3 <id:1..32>
```

Note: the X.25 layer-3 template will not be deleted if it is used by an X.31 interface.

14.3.1.3 X.25 D channel interface (X.31 interface)

An X.31 interface (X.25 over D channel interface) consists of the association of a BRI sub-interface, of a PLL (defined under `permanent-logical-link`) and of an X.25 layer-3 profile (defined under `virtual-`

`template x25-layer3`).

To create an X.31 interface, use the following commands:

```
CLI(configure)> interface x25-dchannel <slot/port.sub-interface>
CLI(conf-if)> permanent-logical-link <pll-number>
CLI(conf-if)> x25-layer-3 profile <id>
CLI(conf-if)> exit
```

`slot/port` represents the BRI interface the D channel of which will carry X.25 data while `sub-interface` is an arbitrary number between 1 and 255 that represents the X.31 interface.

To remove an X.31 interface, use the no form of the above command:

```
CLI(configure)> no interface x25-dchannel <slot/port.sub-interface>
```

Note: removing the X.31 interface does not remove either the PLL or the layer-3 template.

14.3.1.4 X.31 interface routing

The X.25 routing enables to redirect X.25 calls to an output interface especially the X.31 interface.

A call packet cannot be redirected on the interface that it is originating from. The configuration commands follow. To enter in X.25 routing configuration, enter:

```
CLI(configure)> [no] route x25
```

Then select an X.121 address that you want to attach an output X.31 interface afterwards:

```
CLI(route-x25)> [no] x121 <x121-pattern>
```

The pattern entered can be an X.121 number (example: 700011) or a number with a wild card (example: 7000* means all numbers starting with 7000). Any call with an X.121 number matching this entry is routed on the output interface

```
CLI(route-x25)> [no] x25-dchannel <slot/port.sub-interface>
```

Then enter `exit` and create the next X.121 entry if needed. Once you have finished the X.25 routing configuration, enter `execute`.

The routing sorts the entries out from the most discriminating number (longest number without '*') to the least discriminating number ('*'). The routing is made by matching call packets with the sorted entries.

Refer to 14.2.1.4 for more information about X.25 routing.

Example:

```
route x25
 x121 9000*
  x25-dchannel 5/0.1
 exit
 execute
exit
```

14.3.1.5 Statistics and Configuration Display

To display all configuration parameters of the X.31 interface, including default values, use the following commands.

To display the X.25 layer-3 template configuration, use:

```
CLI> show virtual-template x25-layer3 <id>
```

Example:

```
CLI> show virtual-template x25-layer3 1
    PACKET DCE : X121 address = 600000
        no incoming circuits
```

```
         0008 two ways circuits from 0001 to 0008
         no outgoing circuits
         default packet length  128 (input) 128 (output)
         default packet windows 2 (input) 2 (output)
         option [subscribe packet size negociation] is set
         option [subscribe window size negociation] is set
         OUTGOING CALLS OPTIONS :
         option [convert called  address] is not set
         option [remove  called  address] is not set
         option [remove  calling address] is not set
         option [insert CUG] is not set
         no CUG to convert on outgoing call
         INCOMING CALLS OPTIONS :
         option [convert calling address] is not set
         option [insert CUG] is not set
         no CUG to check on incoming call
         no CUG to convert on incoming call
CLI>
```

To display the permanent logical link configuration, use:

```
CLI> show voice voice-port bri { all | index <index> }
```

Example:

```
CLI> show voice voice-port bri index 0
    voice port                              5/0
            protocol descriptor             BRI_TE
            config state                    up
            layer 1 status                  deactivated
            layer 2 status                  TEI , status

            permanent-logical-link          tei , ces , state , status
                    LLP 1                    32 ,   9 ,   -   ,   OFF

            attached voip dial peer         0
            number of voice communication   0
            Channel(s) used
.....................
```

To display the X.31 interface statistics, use the following command:

```
CLI> show statistics x25 [|WORD]

Statistics for X25 D channel interface 5/0.1

Statistics for PACKET LAYER
     [      CALL       ] [      CLEAR      ] [     RESTART     ]
     .start    .lastread .start    .lastread .start    .lastread
Tx :         0         0         0         0         0         0
Rx :         0         0         0         0         0         0
     [    CALL CONF    ] [   CLEAR CONF    ] [  RESTART CONF   ]
     .start    .lastread .start    .lastread .start    .lastread
Tx :         0         0         0         0         0         0
Rx :         0         0         0         0         0         0
     [      DATA       ] [      RESET      ] [    INTERRUPT    ]
     .start    .lastread .start    .lastread .start    .lastread
Tx :         0         0         0         0         0         0
Rx :         0         0         0         0         0         0
     [       RR        ] [   RESET CONF    ] [    INTR.CONF    ]
     .start    .lastread .start    .lastread .start    .lastread
Tx :         0         0         0         0         0         0
Rx :         0         0         0         0         0         0
     [       DIAG      ] [     UNKNOWN     ] [      RNR        ]
     .start    .lastread .start    .lastread .start    .lastread
Rx :         0         0         0         0         0         0
Number of incoming CALL with bad CUG                 0         0
Number of incoming CALL with no CUG                  0         0
```

To see the X.31 active connections status, use the following command:

```
CLI> show x25
+----------------------------------------------------+
| X25 D channel 5/0.1                                |
+----------------------------------------------------+
| Packet Layer is DOWN ( Lower Layer DOWN )          |
| Logical Channels in use 0                          |
+----------------------------------------------------+

+----------------------------------------------------+
| XOT (X25 Over Tcp) interface                       |
+----------------------------------------------------+
| Packet Layer is UP                                 |
| Logical Channels in use 0                         |
+---------------------------------------------------+
```

The X.31 interface status is displayed as follows:

```
CLI> show interfaces x25-dchannel <slot/port.sub-interface>
```

Example:

```
CLI> show interfaces x25-dchannel 5/0.1
x25-dchannel 5/0.1 is allocated, line protocol is started
  Layer 3 template ID 1
  Packet Layer is DOWN ( Lower Layer DOWN )
```

The X.25 routing table is printed as follows:

```
CLI> show x25 route
X25 route : 3 routes
 x121 600010*          routed to XOT
 x121 9000000*         routed to interface x25-dchannel 5/0.1
 x121 90000            routed to XOT
```

14.3.1.6 Debug and Traces

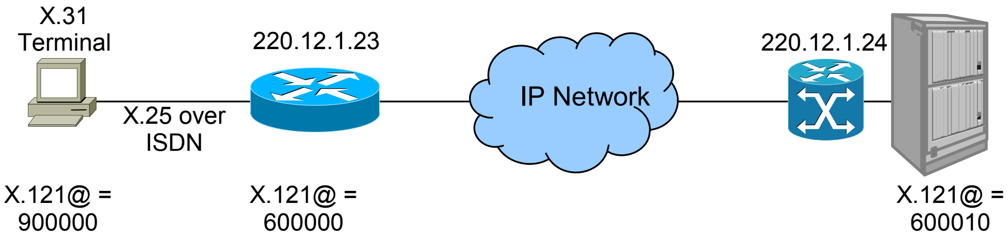
The following command restarts the X31 interface:

```
CLI> x25 restart interface x25-dchannel <slot/port.sub-interface>
```

See 14.2.1.7 for more debug and trace commands.

14.3.1.7 Configuration Example

The following example provides the configuration of an OneOS-based router including an X.31 gateway whose X.121 address is 600000. Its Ethernet interface (220.12.1.23) is connected to an IP network. An X.31 device (X.25 over ISDN) whose X.121 address is 900000 is represented on the left-hand side of the diagram. On the IP network, we find an XOT gateway those IP address is 220.12.1.24 and that manages the X.121 address 600010. See 14.2.1.8 as an XOT example.



```
CLI> show running-config
Building configuration...

Current configuration:

hostname CLI
interface FastEthernet 0/0
 ip address 220.12.1.23 255.255.255.0
exit
virtual-template x25-layer3 1
 packet DCE
 x25-address 600000
exit
interface bri 5/0
 isdn
  application-interface voip
  permanent-logical-link
   llp 1 tei 32
  exit
 exit
 no shutdown
 execute
exit
route xot
 x121 600010*
  ip 220.12.1.24
 exit
 execute
exit
route x25
 x121 600010*
  xot
 exit
 x121 9000000*
  x25-dchannel 5/0.1
 exit
 x121 90000
  xot
 exit
 execute
exit
interface x25-dchannel 5/0.1
 permanent-logical-link 1
 x25-layer3 profile 1
exit
```

14.4 SERIAL COM PORT SERVER

Note: in the following the term "COM Server" is used for "Serial COM Port Server".

The purpose of the COM Server function is to provide an access to the "console" serial port through IP network.

The COM Server function, added in OneOS, is working on a TCP/IP socket. It gets data from the socket and sends it through the serial port and conversely gets data from the serial port and sends it to the socket. Adding to that the serial line configuration can be done by the COM Server thanks to RFC 2217 protocol.

14.4.1 Serial COM Port Server License

COM Server functionalities are only available when "com-port-server" software license is activated. The related commands are rejected if the license is not activated on the router. License activation is possible if the product has sufficiently hardware resources and is pre-loaded with the rights to use the license.

Use the following command in global configuration mode to activate the "com-port-server" license.

```
CLI(configure)> license activate com-port-server
-------------------------------------------------------------
End-User License Agreement
-------------------------------------------------------------
You are only allowed to use the features covered by this
license, if OneAccess has granted permissions to you or your
company.
This command activate the com-port-server command.
By using one of the above mentioned features, you acknowledge
that you got these permissions.
-------------------------------------------------------------
‘com-port-server’ license successfully enabled.

CLI(configure)>
```

Note: the "com-port-server" software license is also activated when the Advanced Security and IP (ASIP) license is activated.

14.4.2 Serial cable handling

Usually, the console serial port is used with a "console" cable connected to a PC for router configuration and management. COM Server function uses a special "serial" cable. This cable can be detected thanks to special unused pin wiring. When the "serial" cable is plugged, a "1" is read by OneOS.

When the "console" cable is plugged or no cable is plugged, a "0" is read by OneOS. The detection of the "serial" cable occurs at start-up or when the device is already started.

14.4.2.1 At start-up

At start-up the checking is done as early as in boot software.

When "serial" cable is detected, no console information is sent by the boot software through the serial port. The operational software will neither send console information through the serial port. When "serial" cable is not detected, the device continues its starting as usual.

Note: because of that, the COM Server function requires a special boot software release.

14.4.2.2 When the device is running

When the device is running, polling is done for "serial" cable detection.

14.4.2.2.1 On "serial" cable detection

All special stuff for "console mode use" is removed. It means:

- The "console mode" parameters are saved.
- The console is disabled (for CLI or tshell that are still remotely available).
- All outputs (including logging) are redirected to the null device.

Then,

- The control signals (usually CTS/RTS) are enabled.
- The "serial mode" parameters are set.
- The "serial-cable-available" flag is set.

14.4.2.2.2 On "serial" cable disconnection

- The "serial-cable-available" flag is unset.
- The "serial mode" parameters are saved.
- The "console mode" parameters are restored.
- The control signals (usually CTS/RTS) are disabled.
- All outputs (including logging) are directed to standard device.
- The console is enabled (CLI or tshell).

14.4.3 TCP connection

When COM Server is not in the "no shutdown" state, it listens to the CLI-configured TCP port.

When a TCP connection request comes in, the source address is checked against the CLI-configured interface (bind) and the CLI-configured access-list.

If this source address is allowed, COM Server accepts the connection and establishes the link between the TCP socket and the serial port. If RFC 2217 mode is used, the serial port "serial mode" parameters are updated and set in the device.

TCP connection is allowed even if no "serial" cable is present. In such case, data coming from the socket are discarded until "serial" cable detection occurs. Conversely, a "serial" cable disconnection does not close the TCP link. Data received from the socket are discarded until the "serial" cable is available again.

The TCP link has to be disconnected by explicit request from the client side.

Note: when a new TCP connection occurs while TCP is already connected, the old connection is closed and the new TCP connection is linked to the serial port.

14.4.4 Software architecture

COM Server is started before configuration file `bsaStart.cfg` execution. It wakes up (polling period 500 ms) only to check "serial" cable detection.

When `bsaStart.cfg` is run, for each `com-port-server` object in `no shutdown` state, the COM Server creates a new socket and starts listening for incoming connection on these TCP ports.

When a `com-port-server` object is set to `shutdown` state, the according listening socket is closed.

Note: Currently, only one serial port is available on OneOS-based routers.

The COM Server is always started even if all `com-port-server` objects are in `shutdown` state in order to have "serial" cable detection.

14.4.5 COM Server configuration

Use the following command in global configuration mode to enter the COM Server configuration:

```
CLI> configure terminal
CLI(configure)> com-port-server <index>
CLI(com-port-server)>
```

Note: As of V4.3 software release, only the COM Server 0 can be activated.

The COM Server must be in the shutdown state (default state) for any change in its configuration:

```
CLI(com-port-server)> shutdown
```

Use the following command, in COM Server configuration mode to define the serial port used:

```
CLI(com-port-server)> hw-port 0
```

Note: As of V4.3 software release, only the CONSOLE port (serial port 0) can be used.

Use the following command to define the TCP port used (6001 by default):

```
CLI(com-port-server)> tcp-port <6000..10000>
```

By default, the COM Server can be accessed through any declared IP address. To restrict access to the IP address of a specific IP interface, use the following command (default any):

```
CLI(com-port-server)> [no] bind <interface/unit[.sub]>
```

This command can be entered several times. Use the no form to remove one interface from the list.

To return to the default settings (COM Server accessed from any interface), enter the following command:

```
CLI(com-port-server)> bind any
```

It is possible to restrict access to the COM server by using a list of addresses standing for the list of permitted/denied source IP addresses (existing access-list). Use the following command:

```
CLI(com-port-server)> [no] acl <acl-name>
```

To save bandwidth in the IP network, characters received from the serial port are put in an IP packet either when the receive buffer is full or when the receive timer elapses.

Note that there is a tradeoff between the saved bandwidth and the delay added by packetization. The more the bandwidth is saved the longer the delay will be.

Use the following command to define the size in characters (default 128) of the receive buffer (IP payload size).

```
CLI(com-port-server)> receive-buffer-size <1..1024>
```

Use the following command to define the duration of the receive timer.

```
CLI(com-port-server)> receive-timer <3..100>
```

The receive timer (default value 10) is expressed in "character duration equivalent". That means that timer duration depends on rate and number of bits per character.

For example: at 9600 bps and 10 bits per character (8N1), a receive timer of 10 means that the next character must be received within 10 ms whereas at 19200 bps, a receive timer of 10 means that the next character must be received within 5 ms.

Note that because the receive timer is restarted each time a character is received, the first character received in the receive buffer may actually be sent after a much longer delay than the receive timer.

Use the following command to define the protocol used (default rfc2217):

```
CLI(com-port-server)> protocol { raw | rfc2217 }
                       ctrl-out { dtr | rts | dtr,rts }
                       ctrl-in { dsr | cts | dcd | ri | dsr,cts | … }
```

`rfc2217`: COM server accepts connection from RFC 2217, telnet or raw client and answers to RFC 2217 or telnet commands.

`raw`: all received chars are transmitted on serial COM port. If RFC 2217 or telnet commands are received they are not handled and are transmitted on serial COM port.

When connection is established using RFC 2217:

`ctrl-out` defines the RFC 2217 outbound control signal: `dtr` or `rts` or `dtr|rts` (default `rts`)

`ctrl-in` defines the RFC 2217 inbound control signal: `dsr` or `cts` or `dcd` or `ri` or part of them or all of them (default `cts`)

Use the following commands to define the serial port parameters at connection startup. In telnet or raw mode these values are not modified afterwards. In RFC 2217 mode these values can be modified at any time during connection by any application setting COM port parameters. New settings are transmitted to COM server using RFC 2217 protocol.

To define the baud rate (default 9600):

```
CLI(com-port-server)> rate { 300 | 600 | 1200 | 2400 | 4800 | 9600
                             | 19200 | 38400 | 57600 | 115200 }
```

To define the number of data bits (default 8):

```
CLI(com-port-server)> data-bit { 5 | 6 | 7 | 8 }
```

To define the parity (default none):

```
CLI(com-port-server)> parity { none | even | odd }
```

To define the number of stop bits (default 1):

```
CLI(com-port-server)> stop-bit { 1 | 2 }
```

To define the flow control (default none):

```
CLI(com-port-server)> flow-control { none | hardware }
```

`hardware` flow control uses inbound control signal. According to the cable, the COM port can be DTE or DCE. If the device is DCE, according to the cable, inbound control signal can be `rts`, `dtr` or both of them. If the device is DTE, according to the cable, inbound control signal can be `cts`, `dsr`, `ri`, `dcd`, part of them or all of them.

To define the outbound control signal management:

```
CLI(com-port-server)> ctrl-out { off | on | cnx | toggle }
```

The outbound control signal can be always open (`off`), always closed (`on`), closed only while COM port is open (`cnx` default value) or open while transmit buffer is empty, closed as soon as transmit buffer is not empty (`toggle`). According to the cable, the COM port can be DTE or DCE. If the device is DTE, according to the cable, outbound control signal can be `rts`, `dtr` or both of them. If the device is DCE, according to the cable, outbound control signal can be `cts`, `dsr`, `ri`, `dcd`, part of them or all of them.

Use the following commands to activate the COM Server:

```
CLI(com-port-server)> no shutdown
CLI(com-port-server)> exit
CLI(configure)> exit
CLI>
```

14.4.6 COM Server Configuration Display

Current configuration can be displayed using the following command in COM Server configuration mode:

```
CLI(com-port-server)> display complete
com-port-server 0
  no acl
  bind any
  ctrl-out cnx
  data-bits 8
  flow-control none
  hw-port 0
  parity none
  protocol rfc2217 ctrl-out rts ctrl-in cts
  rate 9600
  receive-buffer-size 128
  receive-timer 10
  stop-bits 1
  tcp-port 6001
  shutdown
exit
```

Note: in the next 3 sections, because CONSOLE port is used by COM Server, commands and results are achieved through TELNET with the help of "`monitor trace`" command.

14.4.7 COM Server status display

To display COM Server status, use the following command:

```
CLI# show com-port-server status [index <index>]
```

The output of this command depends on cable presence and serial link connection status.

With no "serial" cable presence:

```
CLI# show com-port-server status

SCP server 0
        admin status: no shut
        serial cable detected: no
        oper status: not connected
```

With "serial" cable presence and serial link not connected:

```
CLI# show com-port-server status

SCP server 0
        admin status: no shut
        serial cable detected: yes
        ctrl-out: off
        ctrl-in: on
        oper status: not connected
```

With "serial" cable presence and serial link connected:

```
CLI# show com-port-server status

SCP server 0
        admin status: no shut
        serial cable detected: yes
        ctrl-out: on
        ctrl-in: on
        oper status: connected (rfc2217)
        client address: 10.3.2.1
        rate: 115200 bit/s
        data-bits: 8 bit/char
        parity: none
        stop-bits: 1
```

14.4.8 Statistics Display

To display serial link and TCP socket statistics of the current connection, use the following command:

```
CLI# show com-port-server stats [index <index>]

Current connection:
Serial-port: TX=6 bytes, TX-DISCARD=0 bytes, RX=422 bytes
Tcp-port:    TX=422 bytes, TX-DISCARD=0 bytes, RX=6 bytes
```

To reset current statistics counters, use:

```
CLI# show com-port-server stats reset [index <index>]
```

To display serial link and TCP socket statistics of current and last 4 connections, use:

```
CLI# show com-port-server stats history [index <index>]
```

To reset COM server history counters, use:

```
CLI# show com-port-server stats history reset [index <index>]
```

14.4.9 Debug and Traces

To activate COM Server general traces, use the following command:

```
CLI# trace filter add vox scp scp <detail: 1-4> { mem | log | syslog }
```

To start spying data sent between TCP socket and serial port hardware, use the following command:

```
CLI# debug com-port-server spy on [both|rx|tx] [ascii|hex]
                                               [index <index>]
```

By default both ways are spied in ASCII format. Non printable characters are replaced by space char.

Example: Let's suppose that the following messages are sent and received with the terminal:

```
at&v
DTE channel 10 (Asynchronous V.120) parameters:
 E1 Q0 V1 W0 X4 \Q3 %C0
 &C1 &D2 &K3 &M0 &O10 &Q0 &S0 &Y0
 S000:000 S001:000 S002:043 S003:013 S004:010 S005:008 S007:060
 S012:050 S025:000 S039:003 S048:007 S095:000
 S110:010 S111:008 S112:000 S113:025 S114:115 S116:000 S117:000
 S119:000 S120:000 S121:000 S122:254 S123:000 S124:001 S126:000
 S131:000 S133:080 S153:000 S154:000 S161:000
 &Z00=
OK
```

Then the trace will look as follows (non printable characters are replaced by space chars):

```
CLI# debug com-port-server spy on
01:07:30.821 [TX]a
01:07:30.839 [RX]a
01:07:32.121 [TX]t
01:07:32.140 [RX]t
01:07:34.713 [TX]&
01:07:34.719 [RX]&
01:07:35.706 [TX]v
01:07:35.720 [RX]v
01:07:37.465 [TX]
01:07:37.486 [RX]   DTE channel 10 (Asynchronous V.120) parameters:   E1 Q0 V1 W0 X4 \Q3 %C0
&C1 &D2 &K3 &M0 &O10 &Q0 &S0 &Y0
S000:000 S001
01:07:37.497  [RX]:000  S002:043  S003:013  S004:010  S005:008  S007:060     S012:050  S025:000
S039:003 S048:007 S095:000   S110:010 S111:00
8 S112:000 S1
01:07:37.508  [RX]13:025  S114:115  S116:000  S117:000     S119:000  S120:000  S121:000  S122:254
S123:000 S124:001 S126:000   S131:000 S133:
080 S153:000
01:07:37.512 [RX]S154:000 S161:000   &Z00=    OK
```

To stop spying data:

```
CLI# debug com-port-server spy off
```

To display COM Server events and /or RFC 2217 messages, use the following command:

```
CLI# debug com-port-server info on { gen | rfc2217 | gen,rfc2217 }
```

To stop displaying COM Server events and /or RFC 2217 messages, use the following command:

```
CLI# debug com-port-server info off { gen | rfc2217 | gen,rfc2217 }
```

14.4.10 Configuration Example

The following example removes the default protocol RFC 2217 to force the bitrate to 19200 bps.

```
configure terminal
  com-port-server 0
    protocol raw
    rate 19200
    no shutdown
    exit
  exit
```